

BEA 2026

**The 21st Workshop on Innovative Use of NLP for Building
Educational Applications**

Proceedings of the Workshop

July 3-4, 2026

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

Gold Level



©2026 Association for Computational Linguistics

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 979-8-89176-409-5

Introduction

This year marks the 21st edition of the *Workshop on Innovative Use of NLP for Building Educational Applications*. As usual, we are happy to welcome papers on various aspects and types of educational applications – from traditionally popular tasks around language learning to novel applications related to teaching math and programming languages. This year, BEA continues being a 2-day workshop, with one significant change: we have one hybrid day and one fully virtual day. This change did not only allow us to accept more valuable work from our community, but also ensured better visibility for authors presenting online.

This year, we have received a total of 132 submissions, and from these, we have accepted 19 papers as talks and 44 as poster and demo presentations, for an overall acceptance rate of 48 percent. As in previous years, we have put the main emphasis on the high quality of research when selecting the papers to be accepted, but we also hope that we have managed to bring together a diverse program, with various languages, subject domains, and educational applications covered.

In addition to the diverse oral, poster and demo presentations, this year, we are hosting a panel on *Transitioning from Academia to the EdTech Industry*, with four distinguished speakers involved: Christine Bagarino (Social AI), Kai North (Cambium Assessment), Keelan Evanini (NBME), and Mariano Felice (British Council). BEA 2026 will also include, for the second time, a half-day tutorial on *Theory of Mind and Application in Educational Context*, organized by Effat Farhana (Auburn University), Maha Zainab (Auburn University), Qiaosi Wang (Carnegie Mellon University), Niloofar Miresghallah (Carnegie Mellon University), Ramira van der Meulen (Leiden University), Max van Duijn (Leiden University). Last but not least, this year we have also hosted two shared tasks – on *Vocabulary Difficulty Prediction for English Learners* by Mariano Felice (British Council) and Lucy Skidmore (British Council), and on *Rubric-based Short Answer Scoring for German* by Sebastian Gombert (DIPF), Zhifan Sun (DIPF), Fabian Zehner (DIPF), Jannik Lossjew (IPN), Tobias Wyrwich (IPN), Berrit Katharina Czinczel (IPN), David Bednorz (IPN), Sascha Bernholt (IPN), Knut Neumann (IPN), Ute Harms (IPN), Aiso Heinze (IPN), and Hendrik Drachslar (DIPF). Both tasks attracted a large number of participants, and the program includes oral presentations on the shared tasks from the organizers as well as talks from the tasks' winners and extended poster sessions for shared tasks participants presenting their systems.

Finally, we would like to thank everyone who has been involved in organizing the BEA workshop this year. We are particularly grateful to our sponsors who keep providing their support to BEA: this year, our sponsors include Cambridge University Press & Assessment, Duolingo English Test, Squirrel Ai Learning, CATALPA, and British Council.

BEA 2026 Organizing Committee

Organizing Committee

General Chair

Ekaterina Kochmar, MBZUAI / University of Victoria

Program Chairs

Andrea Horbach, CAU Kiel / Leibniz Institute for Science and Mathematics Education

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

Bashar Alhafni, MBZUAI

Sponsorship Chairs

Zheng Yuan, University of Sheffield

Jill Burstein, Duolingo

Virtual Chair

Stefano Bannò, Cambridge University

Program Committee

Chairs

Bashar Alhafni, MBZUAI
Stefano Bannò, University of Cambridge
Marie Bexte, FernUniversität in Hagen
Jill Burstein, Duolingo
Andrea Horbach, CAU Kiel / Leibniz Institute for Science and Mathematics Education
Ekaterina Kochmar, MBZUAI / University of Victoria
Ronja Laarmann-Quante, Ruhr University Bochum
Anais Tack, KU Leuven; imec
Victoria Yaneva, National Board of Medical Examiners
Zheng Yuan, University of Sheffield

Program Committee

Rania Abdelghani, Hector Institute of Educational Sciences and Psychology, University of Tübingen
Tazin Afrin, NBME
Soroosh Akef, Leibniz-Institut für Wissensmedien
Syeda Sabrina Akter, George Mason University
Vennela Akula, Auburn University
Erfan Al-Hossami, University of North Carolina at Charlotte
Ali Al-Laith, University of Copenhagen
Giora Alexandron, Weizmann Institute of Science
David Alfter, Gothenburg University
Norah Almousa, University of Pittsburgh
Aitor Arronte Alvarez, University of Hawaii at Manoa
Nischal Ashok Kumar, University of Massachusetts Amherst
Sina Bagheri Nezhad, Independent Researcher
Xiaoyu Bai, University of Potsdam
Mohmaed Basem, MSA University
Michael Gringo Angelo Bayona, Trinity College Dublin
Lee Becker, Pearson
Beata Beigman Klebanov, Educational Testing Service
Milena Belosevic, Bielefeld University
Enrico Benedetti, Utrecht University
Luca Benedetto, Telecom SudParis, IP Paris
Maryam Berijanlian, Michigan State University
Ummugul Bezirhan, Boston College, TIMSS and PIRLS International Study Center
Abhidip Bhattacharyya, University of Massachusetts, Amherst
Ted Briscoe, MBZUAI
Dominique Brunato, Institute of Computational Linguistics A. Zampolli (ILC-CNR), Pisa
Andrew Caines, University of Cambridge
Jie Cao, University of Oklahoma
Dumitru-Clementin Cercel, University Politehnica of Bucharest
Bharathi Raja Chakravarthi, University of Galway, Ireland
Imran Chamieh, Hochschule Ruhr West
Ignatios Charalampidis, University of Tuebingen
Andreas Chari, University of Glasgow

Mei-Hua Chen, Department of Foreign Languages and Literature, Tunghai University
 Artem Chernodub, Zendesk
 Luis Chiruzzo, Universidad de la Republica
 Hyundong Cho, USC, Information Sciences Institute
 Declan Clowry, Social AI
 Longwei Cong, DIPF | Leibniz Institute for Research and Information in Education
 Yan Cong, Purdue University
 Mark Core, University of Southern California
 Steven Coyne, Tohoku University / RIKEN
 Orphee De Clercq, LT3, Ghent University
 Kordula De Kuthy, IWM Tübingen
 Jasper Degraeuwe, Ghent University
 Carrie Demmans Epp, University of Alberta
 Yuning Ding, Leibniz Institute for Science and Mathematics Education
 Rahul Divekar, Bentley University
 George Duenas, Universidad Pedagogica Nacional
 Marius Dumitran, University of Bucharest
 Andreea Dutulescu, National University of Science and Technology Politehnica Bucharest
 Yo Ehara, Tokyo Gakugei University
 Walid El Hefny, Leibniz-Institut für Wissensmedien (IWM)
 Effat Farhana, Auburn University
 Mariano Felice, British Council
 Michael Flor, Educational Testing Service
 Jennifer-Carmen Frey, EURAC Research
 Momoka Furuhashi, Tohoku University / National Institute of Informatics
 Thomas Gaillat, Rennes 2 university
 Martina Galletti, Sony Computer Science Laboratories - Paris | Sapienza University of Rome
 Diana Galvan-Sosa, University of Cambridge
 Ashwinkumar Ganesan, Amazon Alexa AI
 Lingyu Gao, Duolingo
 Jie Gao, McGill University
 Rujun Gao, Texas A&M University
 Ritik Garg, Extramarks Education Pvt. Ltd.
 Voula Giouli, Aristotle University of Thessaloniki / ILSP, ATHENA RC
 Aron Gohr, Independent
 Sebastian Gombert, DIPF | Leibniz Institute for Research and Information in Education
 Kiel Gonzales, University of the Philippines Diliman
 Mark Edward Gonzales, Agency for Science, Technology and Research (A*STAR), Singapore
 Cyril Goutte, National Research Council Canada
 Stuart Grey, University of Glasgow
 Viktor Gulbrandsen, University of Oslo
 Pranav Gupta, Cisco
 Abigail Gurin Schleifer, Weizmann Institute of Science
 Ibrahim Hallac, Østfold University of Applied Sciences
 Ching Nam Hang, Assistant Professor, Yam Pak Charitable Foundation School of Computing and Information Sciences, Saint Francis University, Hong Kong
 Jiangang Hao, Educational Testing Service
 Hasnain Heickal, University of Massachusetts Amherst
 Nicolas Hernandez, Nantes University
 Jingying Hu, Purdue University
 Chung-Chi Huang, Frostburg State University

Aiden Huang, Acton-Boxborough Regional High School
 Phoebe Huang, University of Florida
 Anna Huelasing, CAU
 Catherine Ikae, Applied Machine Intelligence, Bern University of Applied Sciences, Switzerland
 Fareya Ikram, University of Massachusetts Amherst
 Joseph Marvin Imperial, University of Bath
 Radu Tudor Ionescu, University of Bucharest
 Md. Rakibul Islam, Ahsanullah University of Science and Technology
 Elsayed Issa, Purdue University
 Raunak Jain, Intuit
 Abhishek Jariwala, Auburn University
 Helen Jin, University of Pennsylvania
 Léane Jourdan, Nantes University
 Indika Kahanda, University of North Florida
 Tomoyuki Kajiwara, Ehime University / The University of Osaka
 Anisia Katinskaia, Swiss Data Science Center
 Fatemeh Kazemi Vanhari, McMaster University
 Elma Kerz, Exaia Technologies
 Samin Khan, Stanford University
 Sachiko Kimura, Rikkyo University, Dokkyo University
 Mamoru Komachi, Hitotsubashi University
 Maxim Konca, Goethe University Frankfurt
 Katsunori Kotani, Kansai Gaidai University
 Joni Kruijsbergen, LT3, Ghent University
 Andrei Kucharavy, HES-SO Valais-Wallis
 Aayush Kucheria, Aalto University
 Roland Kuhn, National Research Council of Canada
 Gaurav Kumar, University of California San Diego
 Ravi Kumar, Florida International University (FIU)
 Alexander Kwako, University of California, Los Angeles
 Antonio Laverghetta Jr., Pennsylvania State University
 Jaewook Lee, UMass Amherst
 Seolhwa Lee, Technical University of Darmstadt
 Bernardo Leite, Faculty of Engineering - University of Porto
 Arun Balajiee Lekshmi Narayanan, University of Pittsburgh
 Hannah Levin, Stanford University
 Jiazheng Li, King's College London
 Xu Li, Southwest Petroleum University
 Junhong Liang, MBZUAI
 Wen Liang, Columbia University
 Chuan-Jie Lin, National Taiwan Ocean University
 Yudong Liu, Western Washington University
 Zhexiong Liu, University of Pittsburgh
 Denise Loefflad, Leibniz-Institut fuer Wissensmedien Tuebingen
 Julian Lohmann, Christian Albrechts Universität Kiel
 Benny Longwill, Educational Testing Service
 Anastassia Loukina, Grammarly Inc
 Wanning Lu, Stanford University
 Li Lucy, University of Washington
 Sarah Löber, University of Tübingen
 Jakub Macina, ETH Zurich

Nitin Madnani, Duolingo
 S.M Afridi Mahmud, Bangladesh University of Engineering and Technology
 Maria Monica Manlises, De La Salle University
 Zhenjiang Mao, University of Florida
 Jacek Marciniak, Adam Mickiewicz University
 Arianna Masciolini, Språkbanken Text; Department of Swedish, Multilingualism, Language Technology; University of Gothenburg
 Sandeep Mathias, Presidency University
 Hunter McNichols, University of Massachusetts Amherst
 Detmar Meurers, Leibniz-Institut für Wissensmedien (IWM)
 Daniel Mora Melanchthon, CAU Kiel / Leibniz Institute for Science and Mathematics Education
 Phoebe Mulcaire, Duolingo
 Laura Musto, Universidad de la Republica
 Farah Nadeem, LUMS
 Sungjin Nam, ACT, Inc
 Diane Napolitano, The Washington Post
 Léo Nebel, LIP6 - Sorbonne Université
 Seyed Parsa Neshaei, EPFL
 Duy Anh Nguyen, Independent Researcher
 Huy Nguyen, Oracle
 S Jaya Nirmala, National Institute of Technology Tiruchirappalli
 Sergiu Nisioi, Human Language Technologies Research Center, University of Bucharest
 Adam Nohejl, RIKEN
 Kai North, Cambium Assessment Inc.
 Similoluwa Okunowo, AIMS South Africa
 Kostiantyn Omelianchuk, Grammarly
 Amin Omidvar, PhD student at the Department of Electrical Engineering and Computer Science, York University
 Daniel Oyeniran, University of Alabama
 Alonso Palomino, DFKI (German Center for Artificial Intelligence Research)
 Nisarg Parikh, University of Massachusetts, Amherst
 Jeiyoon Park, Smilegate
 Kseniia Petukhova, MBZUAI
 Tomasz Pilka, Adam Mickiewicz University in Poznan
 Pagnarith Pit, University of Melbourne
 Lucie Polakova, Charles University
 Long Qin, Alibaba
 Mengyang Qiu, Saint Elizabeth University
 Chatrine Qwaider, MBZUAI
 Md. Abdur Rahman, Islamic University of Technology (IUT), OIC
 Pranil Raichura, Independent Researcher
 Anmol Rao, Arizona State University
 pranshu rastogi, Independent Researcher
 Sparsh Rastogi, Thapar Institute of Engineering and Technology
 Manav Rathod, University of California, Berkeley
 Hanumant Redkar, Goa University, Goa
 Saed Rezayi, National Board of Medical Examiners
 Luisa Ribeiro-Flucht, University of Tuebingen
 Frankie Robertson, University of Jyväskylä
 Shadman Rohan, Center for Computational & Data Sciences, IUB
 Aiala Rosá, Instituto de Computación, Facultad de Ingeniería, Universidad de la República

Mahule Roy, University of Oxford
 Alla Rozovskaya, Queens College, City University of New York
 Josef Ruppenhofer, Fernuniversität in Hagen
 Mariam Saeed, Applied Innovation Center
 Pravish Sainath, Vanier College, Université de Montréal
 Jonathan Sakunkoo, University of Oxford
 Alexander Scarlatos, University of Massachusetts Amherst
 Nils-Jonathan Schaller, Leibniz Institute for Science and Mathematics Education
 Stephanie Schoch, University of Virginia
 Dhruv Shah, Sardar Patel Institute of Technology
 Natarajan Balaji Shankar, University of California Los Angeles
 Matthew Shardlow, Manchester Metropolitan University
 Qian Shen, University of Florida
 Gyu-Ho Shin, University of Illinois Chicago
 Hyo Jeong Shin, Sogang University
 Ziqi Shu, Stanford University
 Astha Singh, Iowa State University
 Li Siyan, Columbia University
 Lucy Skidmore, British Council
 Anastasia Smirnova, San Francisco State University
 Mariia Soliar, Leibniz-Institut für Wissensmedien (IWM)
 Alexey Sorokin, Moscow State University
 Shankhalika Srikanth, Independent Researcher
 Felix Stahlberg, Google Research
 Katherine Stasaski, Google DeepMind
 Helmer Strik, Centre for Language and Speech Technology (CLST), Centre for Language Studies (CLS), Radboud University Nijmegen
 Parisa Suchdev, University of Vermont
 Hakyung Sung, Rochester Institute of Technology
 Andreas Säuberli, LMU Munich
 Kyosuke Takami, Osaka Kyoiku University
 Wenjia Tan, University of Macau
 CheeWei Tan, Nanyang Technological University
 Takehito Utsuro, University of Tsukuba
 Martin Vainikko, University of Tartu
 Sowmya Vajjala, National Research Council
 Piper Vasicek, Brigham Young University
 Justin Vasselli, Nara Institute of Science and Technology
 Hariram Veeramani, UCLA
 Giulia Venturi, Institute of Computational Linguistics Antonio Zampolli (ILC-CNR)
 Anthony Verardi, Duolingo
 Elena Volodina, University of Gothenburg
 Anh-Duc Vu, University of Helsinki
 Deliang Wang, The University of Hong Kong
 Nikhil Wani, OpenThreads AI / University of Southern California
 Taro Watanabe, Nara Institute of Science and Technology
 Alistair Willis, The Open University
 Yiheng Wu, University of Helsinki
 Bushi Xiao, University of Florida
 Hiroaki Yamada, Tokyo Metropolitan University
 Bi-Cheng Yan, National Taiwan Normal University

Haiyin Yang, University of Florida
Roman Yangarber, University of Helsinki
Sahar Yarmohammadtoosky, NBME
Tahreem Yasir, North Carolina State University
Su-Youn Yoon, EduLab
Marcos Zampieri, George Mason University
Fabian Zehner, DIPF | Leibniz Institute for Research and Information in Education
Torsten Zesch, Computational Linguistics, FernUniversität in Hagen
Jing Zhang, Emory University
Yiling Zhao, Stanford University
Yang Zhong, University of Pittsburgh
Ej Zhou, University of Cambridge
Yiyun Zhou, NBME
Jessica Zipf, University of Konstanz
Michael Zock, CNRS-LIS
Robert Östling, Department of Linguistics, Stockholm University

Keynote Talk

Transitioning from Academia to the EdTech Industry

Christine Bagarino
Social AI

Bio: Christine Bagarino is the founder of Social AI, an AI native startup based in Tokyo, Japan, that builds AI technology for positive social impact. A former educator, curriculum developer, and Silicon Valley product manager, Christine developed Japan's first LLM-powered English learning chatbot approved for K12 public education. In 2023, she traveled to rural Vietnam for an EdTech pilot funded by the Asian Development Bank, which inspired her to make fun and quality English learning accessible to children all over the world.

Keynote Talk

Transitioning from Academia to the EdTech Industry

Kai North
Cambium Assessment

Bio: Dr. Kai North is a Senior Data Scientist at Cambium Assessment Inc. Kai earned his PhD in Information Technology at George Mason University, Virginia. He specializes in machine learning and natural language processing applied to educational and assessment technologies and has multiple top-tier publications at venues such as ACL, COLING, and ACM computing surveys.

Keynote Talk

Transitioning from Academia to the EdTech Industry

Keelan Evanini
NBME

Bio: Keelan Evanini has been a researcher in the field of natural language processing (NLP) for educational and conversational applications for more than 15 years. He is currently a Lead NLP Scientist at the National Board of Medical Examiners, where he conducts research into using conversational AI for interactive, scenario-based learning applications for medical students. Prior to joining NBME in 2025, he served as SVP of engineering and AI at Kasisto, a New York-based startup company that provides conversational AI solutions to financial institutions. From 2009-2020, he worked at Educational Testing Service as a research scientist and research director, where he conducted research into automated scoring of non-native spoken English for language proficiency assessment and spoken dialog systems for language learning applications. Keelan received a PhD in linguistics from the University of Pennsylvania in 2009 and is a Senior Member of the IEEE. He has published over 90 peer-reviewed papers and has been awarded 11 US patents.

Keynote Talk

Transitioning from Academia to the EdTech Industry

Mariano Felice
British Council

Bio: Mariano leads the artificial intelligence (AI) strategy for language learning and assessment at the British Council. His role involves researching the application of natural language processing (NLP) to language assessment, providing strategic guidance for the development and adoption of AI solutions, and promoting AI literacy and responsible use of new technologies.

With over a decade of experience, Mariano has worked on a wide range of topics, including grammatical error correction, automatic error typing, system evaluation, automated cloze test generation and item difficulty prediction. He has published numerous scientific papers in top-tier NLP conferences and is a regular speaker at international conferences as well as a reviewer for workshops, journals and conferences in his field. Mariano is also a visiting scholar at the University of Reading and the University of Cambridge.

Table of Contents

Theory of Mind and Application in Educational Context

Effat Farhana, Maha Zainab, Qiaosi Wang, Niloofar Miresghallah, Ramira van der Meulen and Max van Duijn 1

Inferring Student Engagement via Real-Time Thermal–Visual Voice Activity Detection

Bradley Goodman 7

Investigating Context-aware CTC for Pronunciation Assessment: Mitigating Peaky Behavior and Context Independency Assumption

Jiun-Ting Li, Tien-Hong Lo, Bi-Cheng Yan, Shih-Hsuan Chiu, Fu-An Chao and Berlin Chen . 21

A Survey of Automated Presentation Coaching: Systems, Methods, and Open Challenges

Wen Liang, Li Siyan, Zackary Rackauckas and Julia Hirschberg 34

The Aftermath of DrawEduMath: Vision Language Models Underperform with Struggling Students and Misdiagnose Errors

Li Lucy, Albert Zhang, Nathan Anderson, Ryan Knight and Kyle Lo 48

Criterial Features in German: Towards Interpretable NLP in Readability Assessment

Denise Loefflad, Sofia Kathmann, Heiko Holz and Detmar Meurers 63

Letting Tutor Personas Speak Up for LLMs: Learning Steering Vectors from Dialogue via Preference Optimization

Jaewook Lee, Alexander Scarlatos, Simon Woodhead and Andrew Lan 78

Towards Just-in-Time Adaptive Feedback: Enhancing Student Learning via Knowledge-Grounded LLM

Younghun Lee, Amir Bralin, Nobel Sanjay Rebello and Dan Goldwasser 93

RABIT: Rationale-Based Distillation Towards Interpretable Automatic Speaking Assessment via a Small Language Model

Bi-Cheng Yan, Hong-Yun Lin, Fu-An Chao, Jiun-Ting Li and Berlin Chen 108

Towards Pedagogically Aligned LLM Tutors for Math Mistake Remediation

Kseniia Petukhova, Tien Dat Nguyen and Ekaterina Kochmar 118

Challenges in Machine Translation of Interactive Multimodal Exercises

Lucie Polakova, Miroslav Hrabal, Věra Kloudová, Michal Novák, Mariia Anisimova and Martin Popel 141

Evaluating LLM Workflows for Generating Clinical Communication Assessment Items: A Comparative Study with Subject-Matter Experts

Christopher Runyon, Peter Baldwin, Ian Micir, Kevin Frome, Stephanie Mann, Saed Rezayi, Keelan Evanini and Victoria Yaneva 153

Towards Self-Referential Analytic Assessment: A Profile-Based Approach to L2 Writing Evaluation with LLMs

Stefano Banno, Kate Knill and Mark Gales 174

Using Interaction Log Data to Evaluate and Improve Feedback Accuracy in an Intelligent Language Tutoring System

Mariia Soliar, Leona Colling, Stephen Bodnar and Detmar Meurers 190

A Bigger Catch: Fine-Grained Curriculum Standards Alignment on the MathFish Benchmark

Xinman Liu, Mayank Sharma and Xinyu Shi 208

<i>Through the Sentence Lens: Explainable Essay Scoring through Fine-Grained Predictions</i>	
Daniel Mora Melanchthon, Stefan Keller and Andrea Horbach	221
<i>Instruction-Following LLMs for Grammatical Error Correction: Analyzing Neutral-Anchored Instructional Sensitivity Across Editing Modes</i>	
Tolgahan Türker and Gülşen Eryiğit	234
<i>Assessing the Quality and Consistency of Automated Knowledge Component Generation using Instructor-generated Questions and LLMs</i>	
Jordan Esiason, Priyanka Khare, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng and Yeil Jeong	248
<i>Estimating LLM Grading Ability and Response Difficulty in Automatic Short Answer Grading via Item Response Theory</i>	
Longwei Cong, Sonja Hahn, Sebastian Gombert, Leon Camus, Hendrik Drachsler and Ulf Kroe-hne	259
<i>Using k-Shot Prompting with Large k for the Automated Scoring of a German Written Elicited Imitation Test</i>	
Malte Sternik, Ronja Laarmann-Quante and Anastasia Drackert	272
<i>Kelvi: A Morphological Parser to Support Tamil Literacy</i>	
Shankhalika Srikanth, Sabrina Yu, Sophia Chan and Madeline Solis de Ovando	281
<i>From Questions to Assessment Tuples: A Multi-Agent Framework with Bloom-Specialized Agents and Automated Verification</i>	
Gee-Lyle Wong, Runcong Zhao, Yulan He and Jiazheng Li	292
<i>Intent vs. Surface: Recovering Acoustic Realization from Modern ASR for Pronunciation Training</i>	
Seongjin Park	336
<i>KEYSCORE — Keystroke-enhanced Automated Essay Scoring</i>	
Nils-Jonathan Schaller, Daniel Mora Melanchthon, Thorben Jansen, Olaf Köller and Andrea Horbach	347
<i>EduMUSE: A Multimodal Educational Dataset with Automatically Extracted Instructional Context</i>	
Andreea Dutulescu, Stefan Ruseti, Mihai Dascalu and Danielle McNamara	358
<i>Opportunities and Challenges of LLMs in Education: An NLP Perspective</i>	
Sowmya Vajjala, Bashar Alhafni, Stefano Banno, Kaushal Maurya and Ekaterina Kochmar ..	368
<i>Fine-Grained Content Zone Prediction in German Argumentative Essays Using LLMs</i>	
Xiaoyu Bai and Manfred Stede	405
<i>Multi-step Large Language Model for Fine-Grained Feedback in Stepwise Linear Equation Solutions</i>	
Imran Chamieh, Torsten Zesch and Klaus Giebertmann	419
<i>Quality-Conditioned Agreement in Automated Short Answer Scoring: Mid-Range Degradation and the Impact of Task-Specific Adaptation</i>	
Abigail Gurin Schleifer, Moriah Ariely, Beata Beigman Klebanov, Asaf Salman and Giora Alexandron	432
<i>Using LLMs for item creation: Validating the potential of automatically generated sentence repetition test items for language assessment</i>	
Sarah Löber, Björn Rudzewitz, Yuan Chu, Mengyuan He, Shiqin Liu, Yushan Ye and Xiaobin Chen	443

<i>Comparative Evaluation of AI-Generated vs. Expert-written Answer Explanations for a Medical Education Self-Assessment</i>	
Yiyun Zhou, Francis O'Donnell and Victoria Yaneva	455
<i>What Aggregate Scores Hide: Per-Rule Evaluation of Russian Grammatical Error Correction</i>	
Anna Smirnova, Artyom Kopan, Vladislav Makeev and George Chernishev	463
<i>FinnGEC: Benchmarking Grammatical Error Correction for Finnish</i>	
Anh-Duc Vu, Mikhail Zolotilin, Jue Hou, Anisia Katinskaia, Yiheng Wu and Roman Yangarber	479
<i>From Metrics to Meaning: Rule-Grounded LLM Explanations for Data Literacy in the Case of Youth Football</i>	
Tomasz Piłka, Tomasz Kuczyński and Mateusz Czajka	492
<i>Sharing is Caring: Advantages of Sharing a Language Background with Learners as an Annotator of Learner Data in UD</i>	
Caroline Grand-Clement and Arianna Masciolini	503
<i>Classification of Student Struggle in Mathematics</i>	
Hannah Levin, Madhura Padwal and Nchimunya Mwiinga	513
<i>PERSA: Reinforcement Learning for Professor-Style Personalized Feedback with LLMs</i>	
Ravi Kumar, Utkarsh Grover, Xiaomin Lin and Agoritsa Polyzou	529
<i>Data-lean fine-tuning of models for evaluating teacher performance in a GenAI-led elicitation simulation</i>	
Beata Beigman Klebanov, Andrew Hoang, Jamie Mikeska, Benny Longwill, Sanjna Kashyap, Shreyashi Halder and Aakanksha Bhatia	546
<i>Multi-component student writing profiles for expert-aligned automated evaluation of English learner essays.</i>	
Russell Moore, Andrew Caines and Paula Buttery	562
<i>Policy-Sensitive Fairness Evaluation in Automated Scoring of Clinical Communication</i>	
Saed Rezayi, Le An Ha, Victoria Yaneva, Polina Harik, Janet Mee and Jason Snyder	574
<i>Noise Steering for Controlled Text Generation: Improving Diversity and Reading-Level Fidelity in Arabic Educational Story Generation</i>	
Haziq Khalid, Salsabeel Shapsough and Imran Zualkernan	581
<i>The Effects of Structured LLM-Generated Feedback on Programming Assignment Performance</i>	
Tsvetomila Mihaylova, Evanfiya Logacheva, Arto Hellas, Jing Fan, Francisco Castro, Bitá Akram, Narges Norouzi, Peter Brusilovsky and Juho Leinonen	598
<i>Interpretable Difficulty-Aware Knowledge Tracing in Tutor-Student Dialogues</i>	
Shuyan Huang, Alexander Scarlatos, Jaewook Lee and Andrew Lan	612
<i>Rubrics as Semantic Subspaces: A Unified Approach to Rubric-based Constructed Response Scoring across Short Answers and Essays</i>	
Sebastian Gombert, Sonja Hahn, Nico Andersen, Leon Camus, Zhifan Sun, Ngoc Nhu Hao Nguyen, Fabian Zehner, Longwei Cong, Alexander Mehler and Hendrik Drachslér	624
<i>Domain-Adaptive Pre-training for Automated Short Answer Grading in Conceptual Physics: Reliability, Question-Level Analysis, and Error Reduction</i>	
Shirin Lade, Alistair Willis, Jonathan Nylk and Oli Howson	635

<i>Measuring Optimal Challenge: Trajectory-Based Difficulty Alignment in Open-Ended Language Tutoring</i>	
Ziqi Shu, Shuman Wang and Michael Hardy	651
<i>PeerMathDial: A Middle School Dialogue Dataset for Student Collaborative Math Problem Solving</i>	
Murong Yue, Desmond Mcglone, Emily Slutz, Wenhan Lyu, Yixuan Zhang, Jennifer Suh and Ziyu Yao	668
<i>Effects of Varying LLM Access on Essay Writing Behavior</i>	
Julia Christenson, Karin de Langis, Shirley Anugrah Hayati and Dongyeop Kang	685
<i>Assessment of L2 speech global dimensions using large audio language models</i>	
Elsayed Issa and Mahmoud Ali	702
<i>Incentives Of EdTech: A Systematic Review Of EduNLP Research</i>	
Gabrielle Gaudeau, Aoife O'Driscoll, Jasper Degraeuwe, Andrew Caines, Donya Roeein and Zeerak Talat	715
<i>Children's English Reading Story Generation via Supervised Fine-Tuning of Compact LLMs with Controllable Difficulty and Safety</i>	
Qian Shen, Fanghua Cao, Min Yao, Shlok Gilda, Bonnie Dorr and Walter Leite	751
<i>Transformer-based readability classifiers are worse than you think: Evidence from cross-domain Arabic readability assessment</i>	
SARH ALZU'BI and Robert Reynolds	766
<i>Predicting Item Difficulty and Generating Reading Comprehension Items via an Annotated Repository</i>	
Radhika Kapoor, Mayank Sharma, Sang Truong, Nick Haber, Ben Domingue and Maria Ruiz-Primo	777
<i>Generative-Evaluative Agreement: A Necessary Validity Criterion for LLM-Enabled Adaptive Assessment</i>	
Grandee Lee, Yue Wang, Che Yee Lye and Luke Peh	798
<i>Evaluating LLM-Generated Formative Feedback for Undergraduate Mathematics Through the Lens of Feedback Theory</i>	
Aron Gohr, Marie-Amelie Lawn, Kevin Gao, Inigo Serjeant and Stephen Heslip	813
<i>Confirming Correct, Missing the Rest: LLM Tutoring Agents Struggle Where Feedback Matters Most</i>	
Tahreem Yasir, Wenbo Li, Sam Gilson, Sutapa Tithi, Xiaoyi Tian and Tiffany Barnes	819
<i>Retrieval-Augmented Tutoring for Algorithm Tracing and Problem-Solving in AI Education</i>	
Mragisha Jain, Tirth Bhatt, Griffin Pitts, Aum Pandya, Peter Brusilovsky, Narges Norouzi, Arto Hellas, Juho Leinonen and Bitu Akram	841
<i>LLM-Powered but Rule-Grounded: Pedagogically Relevant Grammatical Error Characterization for Learner Model Construction</i>	
Soroosh Akef, Amália Mendes, P Rebuschat and Detmar Meurers	851
<i>Teaching Through Analogies: A Modular Pipeline for Educational Analogy Generation</i>	
Mariam Barakat and Ekaterina Kochmar	863
<i>Edit-level Majority Voting Mitigates Over-Correction in LLM-based Grammatical Error Correction</i>	
Takumi Goto, Yusuke Sakai and Taro Watanabe	899
<i>Zero Shot Phonics: Evaluating Constraint-Adherent Phonics Story Generation in Large Language Models</i>	
Maria Monica Manlises and Ethel Ong	914

<i>From Dialogue to Learner Modeling: Identifying Candidate Signals of Productive Use in LLM-Based Grammar Practice</i>	
Luisa Ribeiro-Flucht, Lanhua Huang and Xiaobin Chen	933
<i>Evaluating Adaptive Personalization of Educational Readings with Simulated Learners</i>	
Ryan Woo, Anmol Rao, Aryan Keluskar and Yinong Chen	941
<i>Toward Cross-Domain Automated Feedback: A Comparative Evaluation of Open-Source Models across Diverse Student Assessment Types</i>	
Muhammad Haseeb, Min Paing Hmue, Ahmad Imam Amjad, Maaz Amjad and Victor Sheng	951
<i>Findings of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners</i>	
Mariano Felice and Lucy Skidmore	964
<i>SATLab at BEA 2026 Shared Task 1: Predicting the Difficulty of English Words for Three L1 Learners Using Primarily Psycholinguistic Features</i>	
Yves Bestgen	985
<i>UGA Threshold at BEA 2026 Shared Task 1: Predicting Vocabulary Acquisition Difficulty with Hand-Crafted SLA-Based Features</i>	
Emma Dalbo	992
<i>TeamXBC at BEA 2026 Shared Task 1: How AI (and I) won the shared task: Vibe and agentic coding solutions for practical machine learning problems</i>	
Xiaobin Chen	997
<i>SAAKTH at BEA 2026 Shared Task 1: L1-Aware English Vocabulary Difficulty Prediction with Hybrid Transformer and Psycholinguistic Features</i>	
Karthik Mattu, Adit Dhall, Arshad Naguru, Shubh Sehgal, Thejas Gowda and Hakyung Sung	1010
<i>SurreyCTS at BEA 2026 Shared Task 1: Semantic Funnelling and Entropy-based Multilingual Lexical Difficulty Prediction</i>	
Georgina Willoughby, Jordan Painter, Diptesh Kanojia, Emily Wells and Constantin Orasan	1016
<i>EduNLP at BEA 2026 Shared Task 1: Multi-Model Ensemble with Feature-Augmented Transformers for Vocabulary Difficulty Prediction</i>	
Avinash Kumar Sharma	1024
<i>AIDA at BEA 2026 Shared Task 1: A Two-Stage Framework for L1-Aware Vocabulary Difficulty Prediction with Representation Diversity and Residual Calibration</i>	
Seok Hyeon Cho, JunHyeok Choi, Sangeun Ji and Sung Won Han	1029
<i>Failure at BEA 2026 Shared Task 1: One Pipeline, Three L1s: A Unified Language-Agnostic System for Vocabulary Difficulty Prediction</i>	
Abid Hossain and Kamruzzaman Khan Alve	1041
<i>BoostedCats at BEA 2026 Shared Task 1: What Makes a Word Hard to Learn? Modeling L1 Influence on English Vocabulary Difficulty</i>	
Jonas Mayer Martins, Zhuojing Huang, Aaricia Herygers and Lisa Beinborn	1047
<i>uogal at BEA 2026 Shared Task 1: Ensemble of Multilingual Encoders with NMT Augmentation for L1-Aware Vocabulary Difficulty Prediction</i>	
bernardo stearns, John P. McCrae, Thomas Gaillat and Jefkine Kafunah	1065
<i>Jinnie's Lab at BEA 2026 Shared Task 1: Precalibration of Vocabulary Item Difficulty with Multilingual Transformers and Multi-Task Learning</i>	
Zhe Li, Pauline Aguinale and Jinnie Shin	1077

<i>Glite at BEA 2026 Shared Task 1: Holistic Difficulty Models Dominate, Feature Engineering Closes the Gap in L1-Aware Vocabulary Difficulty Prediction</i>	
Vassili Philippov, Dmitrii Andreev, Pavel Katunin and Anton Nikolaev	1091
<i>NLP-Explorers at BEA 2026 Shared Task 1: DeBERTa–CatBoost Weighted Ensemble Approach for L1-Specific Vocabulary Difficulty Prediction</i>	
Tayyab Latif, Asifa Bibi, Sabur Butt, Grigori Sidorov and Alexander Gelbukh	1106
<i>RETUYT-INCO at BEA 2026 Shared Task 1: Feature-Enriched mDeBERTa for Word Difficulty Prediction</i>	
Santiago Robaina, Aiala Rosá and Luis Chiruzzo	1113
<i>Token Titans at BEA 2026 Shared Task 1: Multilingual Lexical Complexity Prediction via Fine-Tuned XLM-RoBERTa with Ensemble Decoding</i>	
Anubhab Parashar and Sandeep Mathias	1119
<i>TOEBM at BEA 2026 Shared Task 1: Improving Lexical Difficulty Prediction with Context-Aligned Contrastive Learning and Ridge Ensembling</i>	
wicaksono M., Joanito Lopo, Tsamarah Nugraha, Ahmad Adi and Muhamad Nurfajri	1124
<i>Data Asgardians at BEA 2026 Shared Task 1: A Hybrid Transformer–Feature Ensemble for L1-Aware English Vocabulary Difficulty Prediction</i>	
Adrian Pineda, Sabur Butt and Héctor Ceballos Cancino	1137
<i>UOL@IDEM at BEA 2026 Shared Task 1: Neural Fusion and Feature-Rich Modeling for L1-Aware Vocabulary Difficulty Prediction</i>	
Nouran Khallaf and Serge Sharoff	1146
<i>Sakura at BEA 2026 Shared Task 1: What Makes Vocabulary Difficult?</i>	
Adam Nohejl, Xuanxin Wu, Yusuke Ide, Maria Riera Machin and Yi-Ning Chang	1161
<i>Report on the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German</i>	
Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berit Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze and Hendrik Drachsler	1179
<i>HFT at BEA 2026 Shared Task 2: Blunt-Edge Models for Hybrid Grading</i>	
Ulrike Pado	1193
<i>ASLAN at BEA 2026 Shared Task 2: Voting Across Scoring Paradigms</i>	
Marie Bexte, Yuning Ding, Josef Ruppenhofer, Nils-Jonathan Schaller, Daniel Mora Melanchthon, Torsten Zesch and Andrea Horbach	1201
<i>WSE Research at BEA 2026 Shared Task 2: Multi-Strategy Rubric-Based Short Answer Scoring for German</i>	
Jonas Gwozdz and Andreas Both	1210
<i>AMATI at BEA 2026 Shared Task 2: Automatic Short Answer Grading with Inductive Logic Programming and a Large Language Model</i>	
Alistair Willis and Aisling Third	1217
<i>IWM-DKM at BEA 2026 Shared Task 2: Supplementing Supervised Fine-Tuning for Rubric-Based Short Answer Scoring</i>	
Kate Belcher, Marius De Kuthy Meurers, Kordula De Kuthy and Detmar Meurers	1224
<i>RETUYT-INCO at BEA 2026 Shared Task 2: Meta-prompting in Rubric-based Scoring for German</i>	
Ignacio Sastre, Ignacio Remersaro, Facundo Díaz, Nicolás De Horta, Luis Chiruzzo, Aiala Rosá and Santiago Góngora	1235

<i>SDPA at BEA 2026 Shared Task 2: Efficient LLM Fine-Tuning for Rubric-based Short Answer Scoring</i>	
Zhexiong Liu and Jing Zhang	1244

Program

Friday, July 3, 2026

06:00 - 07:30 *Early Poster Session*

Using k-Shot Prompting with Large k for the Automated Scoring of a German Written Elicited Imitation Test

Malte Sternik, Ronja Laarmann-Quante and Anastasia Drackert

Fine-Grained Content Zone Prediction in German Argumentative Essays Using LLMs

Xiaoyu Bai and Manfred Stede

Sharing is Caring: Advantages of Sharing a Language Background with Learners as an Annotator of Learner Data in UD

Caroline Grand-Clement and Arianna Masciolini

The Effects of Structured LLM-Generated Feedback on Programming Assignment Performance

Tsvetomila Mihaylova, Evanfiya Logacheva, Arto Hellas, Jing Fan, Francisco Castro, Bitá Akram, Narges Norouzi, Peter Brusilovsky and Juho Leinonen

HFT at BEA 2026 Shared Task 2: Blunt-Edge Models for Hybrid Grading

Ulrike Pado

WSE Research at BEA 2026 Shared Task 2: Multi-Strategy Rubric-Based Short Answer Scoring for German

Jonas Gwozdz and Andreas Both

07:30 - 08:45 *Early Oral Session*

Domain-Adaptive Pre-training for Automated Short Answer Grading in Conceptual Physics: Reliability, Question-Level Analysis, and Error Reduction

Shirin Lade, Alistair Willis, Jonathan Nylk and Oli Howson

Using Interaction Log Data to Evaluate and Improve Feedback Accuracy in an Intelligent Language Tutoring System

Mariia Soliar, Leona Colling, Stephen Bodnar and Detmar Meurers

Towards Pedagogically Aligned LLM Tutors for Math Mistake Remediation

Kseniia Petukhova, Tien Dat Nguyen and Ekaterina Kochmar

What Aggregate Scores Hide: Per-Rule Evaluation of Russian Grammatical Error Correction

Anna Smirnova, Artyom Kopan, Vladislav Makeev and George Chernishev

Friday, July 3, 2026 (continued)

IWM-DKM at BEA 2026 Shared Task 2: Supplementing Supervised Fine-Tuning for Rubric-Based Short Answer Scoring

Kate Belcher, Marius De Kuthy Meurers, Kordula De Kuthy and Detmar Meurers

09:00 - 10:30 *Tutorial Session A*

Theory of Mind and Application in Educational Context

Effat Farhana, Maha Zainab, Qiaosi Wang, Niloofar Miresghallah, Ramira van der Meulen and Max van Duijn

10:30 - 11:00 *Coffee Break*

11:00 - 12:30 *Tutorial Session B*

Theory of Mind and Application in Educational Context

Effat Farhana, Maha Zainab, Qiaosi Wang, Niloofar Miresghallah, Ramira van der Meulen and Max van Duijn

12:30 - 14:00 *Lunch Break / Birds of a Feather*

14:00 - 15:45 *Oral Session A*

Using Interaction Log Data to Evaluate and Improve Feedback Accuracy in an Intelligent Language Tutoring System

Mariia Soliar, Leona Colling, Stephen Bodnar and Detmar Meurers

Estimating LLM Grading Ability and Response Difficulty in Automatic Short Answer Grading via Item Response Theory

Longwei Cong, Sonja Hahn, Sebastian Gombert, Leon Camus, Hendrik Drachsler and Ulf Kroehne

LLM-Powered but Rule-Grounded: Pedagogically Relevant Grammatical Error Characterization for Learner Model Construction

Soroosh Akef, Amália Mendes, P Rebuschat and Detmar Meurers

KEYSCORE — Keystroke-enhanced Automated Essay Scoring

Nils-Jonathan Schaller, Daniel Mora Melanchthon, Thorben Jansen, Olaf Köller and Andrea Horbach

15:45 - 16:00 *Coffee Break*

Friday, July 3, 2026 (continued)

16:00 - 17:30 *Poster Session A*

Inferring Student Engagement via Real-Time Thermal–Visual Voice Activity Detection

Bradley Goodman

Letting Tutor Personas Speak Up for LLMs: Learning Steering Vectors from Dialogue via Preference Optimization

Jaewook Lee, Alexander Scarlatos, Simon Woodhead and Andrew Lan

A Bigger Catch: Fine-Grained Curriculum Standards Alignment on the MathFish Benchmark

Xinman Liu, Mayank Sharma and Xinyu Shi

Through the Sentence Lens: Explainable Essay Scoring through Fine-Grained Predictions

Daniel Mora Melanchthon, Stefan Keller and Andrea Horbach

Kelvi: A Morphological Parser to Support Tamil Literacy

Shankhalika Srikanth, Sabrina Yu, Sophia Chan and Madeline Solis de Ovando

From Questions to Assessment Tuples: A Multi-Agent Framework with Bloom-Specialized Agents and Automated Verification

Gee-Lyle Wong, Runcong Zhao, Yulan He and Jiazheng Li

Multi-step Large Language Model for Fine-Grained Feedback in Stepwise Linear Equation Solutions

Imran Chamieh, Torsten Zesch and Klaus Giebertmann

Comparative Evaluation of AI-Generated vs. Expert-written Answer Explanations for a Medical Education Self-Assessment

Yiyun Zhou, Francis O'Donnell and Victoria Yaneva

Multi-component student writing profiles for expert-aligned automated evaluation of English learner essays.

Russell Moore, Andrew Caines and Paula Buttery

Policy-Sensitive Fairness Evaluation in Automated Scoring of Clinical Communication

Saed Rezayi, Le An Ha, Victoria Yaneva, Polina Harik, Janet Mee and Jason Snyder

Friday, July 3, 2026 (continued)

Rubrics as Semantic Subspaces: A Unified Approach to Rubric-based Constructed Response Scoring across Short Answers and Essays

Sebastian Gombert, Sonja Hahn, Nico Andersen, Leon Camus, Zhifan Sun, Ngoc Nhu Hao Nguyen, Fabian Zehner, Longwei Cong, Alexander Mehler and Hendrik Drachslar

Assessment of L2 speech global dimensions using large audio language models

Elsayed Issa and Mahmoud Ali

Children's English Reading Story Generation via Supervised Fine-Tuning of Compact LLMs with Controllable Difficulty and Safety

Qian Shen, Fanghua Cao, Min Yao, Shlok Gilda, Bonnie Dorr and Walter Leite

Transformer-based readability classifiers are worse than you think: Evidence from cross-domain Arabic readability assessment

SARH ALZU'BI and Robert Reynolds

Predicting Item Difficulty and Generating Reading Comprehension Items via an Annotated Repository

Radhika Kapoor, Mayank Sharma, Sang Truong, Nick Haber, Ben Domingue and Maria Ruiz-Primo

Generative-Evaluative Agreement: A Necessary Validity Criterion for LLM-Enabled Adaptive Assessment

Grandee Lee, Yue Wang, Che Yee Lye and Luke Peh

Teaching Through Analogies: A Modular Pipeline for Educational Analogy Generation

Mariam Barakat and Ekaterina Kochmar

From Dialogue to Learner Modeling: Identifying Candidate Signals of Productive Use in LLM-Based Grammar Practice

Luisa Ribeiro-Flucht, Lanhua Huang and Xiaobin Chen

Toward Cross-Domain Automated Feedback: A Comparative Evaluation of Open-Source Models across Diverse Student Assessment Types

Muhammad Haseeb, Min Paing Hmue, Ahmad Imam Amjad, Maaz Amjad and Victor Sheng

SAAKTH at BEA 2026 Shared Task 1: L1-Aware English Vocabulary Difficulty Prediction with Hybrid Transformer and Psycholinguistic Features

Karthik Mattu, Adit Dhall, Arshad Naguru, Shubh Sehgal, Thejas Gowda and Hakyung Sung

SDPA at BEA 2026 Shared Task 2: Efficient LLM Fine-Tuning for Rubric-based Short Answer Scoring

Zhexiong Liu and Jing Zhang

Friday, July 3, 2026 (continued)

*TOEBM at BEA 2026 Shared Task 1: Improving Lexical Difficulty Prediction
with Context-Aligned Contrastive Learning and Ridge Ensembling*

wicaksono M., Joanito Lopo, Tsamarah Nugraha, Ahmad Adi and Muhamad Nur-
fajri

Saturday, July 4, 2026

09:00 - 10:30 *Oral Session B*

The Aftermath of DrawEduMath: Vision Language Models Underperform with Struggling Students and Misdiagnose Errors

Li Lucy, Albert Zhang, Nathan Anderson, Ryan Knight and Kyle Lo

Interpretable Difficulty-Aware Knowledge Tracing in Tutor-Student Dialogues

Shuyan Huang, Alexander Scarlatos, Jaewook Lee and Andrew Lan

Measuring Optimal Challenge: Trajectory-Based Difficulty Alignment in Open-Ended Language Tutoring

Ziqi Shu, Shuman Wang and Michael Hardy

Findings of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners

Mariano Felice and Lucy Skidmore

Sakura at BEA 2026 Shared Task 1: What Makes Vocabulary Difficult?

Adam Nohejl, Xuanxin Wu, Yusuke Ide, Maria Riera Machin and Yi-Ning Chang

Report on the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German

Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berit Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze and Hendrik Drachslar

10:30 - 11:00 *Coffee Break*

11:00 - 12:30 *Oral Session C*

EduMUSE: A Multimodal Educational Dataset with Automatically Extracted Instructional Context

Andreea Dutulescu, Stefan Ruseti, Mihai Dascalu and Danielle McNamara

Confirming Correct, Missing the Rest: LLM Tutoring Agents Struggle Where Feedback Matters Most

Tahreem Yasir, Wenbo Li, Sam Gilson, Sutapa Tithi, Xiaoyi Tian and Tiffany Barnes

Towards Just-in-Time Adaptive Feedback: Enhancing Student Learning via Knowledge-Grounded LLM

Younghun Lee, Amir Bralin, Nobel Sanjay Rebello and Dan Goldwasser

Saturday, July 4, 2026 (continued)

Evaluating LLM Workflows for Generating Clinical Communication Assessment Items: A Comparative Study with Subject-Matter Experts

Christopher Runyon, Peter Baldwin, Ian Micir, Kevin Frome, Stephanie Mann, Saed Rezayi, Keelan Evanini and Victoria Yaneva

Zero Shot Phonics: Evaluating Constraint-Adherent Phonics Story Generation in Large Language Models

Maria Monica Manlises and Ethel Ong

Evaluating Adaptive Personalization of Educational Readings with Simulated Learners

Ryan Woo, Anmol Rao, Aryan Keluskar and Yinong Chen

12:30 - 14:00 *Lunch Break / Birds of a Feather*

14:00 - 15:30 *Poster Session B*

Investigating Context-aware CTC for Pronunciation Assessment: Mitigating Peak Behavior and Context Independency Assumption

Jiun-Ting Li, Tien-Hong Lo, Bi-Cheng Yan, Shih-Hsuan Chiu, Fu-An Chao and Berlin Chen

A Survey of Automated Presentation Coaching: Systems, Methods, and Open Challenges

Wen Liang, Li Siyan, Zackary Rackauckas and Julia Hirschberg

Criterial Features in German: Towards Interpretable NLP in Readability Assessment

Denise Loefflad, Sofia Kathmann, Heiko Holz and Detmar Meurers

RABIT: Rationale-Based Distillation Towards Interpretable Automatic Speaking Assessment via a Small Language Model

Bi-Cheng Yan, Hong-Yun Lin, Fu-An Chao, Jiun-Ting Li and Berlin Chen

Challenges in Machine Translation of Interactive Multimodal Exercises

Lucie Polakova, Miroslav Hrabal, Věra Kloudová, Michal Novák, Mariia Anisimova and Martin Popel

Towards Self-Referential Analytic Assessment: A Profile-Based Approach to L2 Writing Evaluation with LLMs

Stefano Banno, Kate Knill and Mark Gales

Assessing the Quality and Consistency of Automated Knowledge Component Generation using Instructor-generated Questions and LLMs

Jordan Esiason, Priyanka Khare, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng and Yeil Jeong

Saturday, July 4, 2026 (continued)

Intent vs. Surface: Recovering Acoustic Realization from Modern ASR for Pronunciation Training

Seongjin Park

Opportunities and Challenges of LLMs in Education: An NLP Perspective

Sowmya Vajjala, Bashar Alhafni, Stefano Banno, Kaushal Maurya and Ekaterina Kochmar

Quality-Conditioned Agreement in Automated Short Answer Scoring: Mid-Range Degradation and the Impact of Task-Specific Adaptation

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

Using LLMs for item creation: Validating the potential of automatically generated sentence repetition test items for language assessment

Sarah Löber, Björn Rudzewitz, Yuan Chu, Mengyuan He, Shiqin Liu, Yushan Ye and Xiaobin Chen

FinnGEC: Benchmarking Grammatical Error Correction for Finnish

Anh-Duc Vu, Mikhail Zolotilin, Jue Hou, Anisia Katinskaia, Yiheng Wu and Roman Yangarber

From Metrics to Meaning: Rule-Grounded LLM Explanations for Data Literacy in the Case of Youth Football

Tomasz Piłka, Tomasz Kuczyński and Mateusz Czajka

Classification of Student Struggle in Mathematics

Hannah Levin, Madhura Padwal and Nchimunya Mwiinga

PERSA: Reinforcement Learning for Professor-Style Personalized Feedback with LLMs

Ravi Kumar, Utkarsh Grover, Xiaomin Lin and Agoritsa Polyzou

Data-lean fine-tuning of models for evaluating teacher performance in a GenAI-led elicitation simulation

Beata Beigman Klebanov, Andrew Hoang, Jamie Mikeska, Benny Longwill, Sanjna Kashyap, Shreyashi Halder and Aakanksha Bhatia

Noise Steering for Controlled Text Generation: Improving Diversity and Reading-Level Fidelity in Arabic Educational Story Generation

Haziq Khalid, Salsabeel Shapsough and Imran Zualkernan

PeerMathDial: A Middle School Dialogue Dataset for Student Collaborative Math Problem Solving

Murong Yue, Desmond Mcglone, Emily Slutz, Wenhan Lyu, Yixuan Zhang, Jennifer Suh and Ziyu Yao

Saturday, July 4, 2026 (continued)

Evaluating LLM-Generated Formative Feedback for Undergraduate Mathematics Through the Lens of Feedback Theory

Aron Gohr, Marie-Amelie Lawn, Kevin Gao, Inigo Serjeant and Stephen Heslip

Retrieval-Augmented Tutoring for Algorithm Tracing and Problem-Solving in AI Education

Mragisha Jain, Tirth Bhatt, Griffin Pitts, Aum Pandya, Peter Brusilovsky, Narges Norouzi, Arto Hellas, Juho Leinonen and Bitu Akram

Edit-level Majority Voting Mitigates Over-Correction in LLM-based Grammatical Error Correction

Takumi Goto, Yusuke Sakai and Taro Watanabe

AIDA at BEA 2026 Shared Task 1: A Two-Stage Framework for L1-Aware Vocabulary Difficulty Prediction with Representation Diversity and Residual Calibration

Seok Hyeon Cho, JunHyeok Choi, Sangeun Ji and Sung Won Han

BoostedCats at BEA 2026 Shared Task 1: What Makes a Word Hard to Learn? Modeling L1 Influence on English Vocabulary Difficulty

Jonas Mayer Martins, Zhuojing Huang, Aaricia Herygers and Lisa Beinborn

uogal at BEA 2026 Shared Task 1: Ensemble of Multilingual Encoders with NMT Augmentation for L1-Aware Vocabulary Difficulty Prediction

bernardo stearns, John P. McCrae, Thomas Gaillat and Jefkine Kafunah

Jinnie's Lab at BEA 2026 Shared Task 1: Precalibration of Vocabulary Item Difficulty with Multilingual Transformers and Multi-Task Learning

Zhe Li, Pauline Aguinalde and Jinnie Shin

IWM-DKM at BEA 2026 Shared Task 2: Supplementing Supervised Fine-Tuning for Rubric-Based Short Answer Scoring

Kate Belcher, Marius De Kuthy Meurers, Kordula De Kuthy and Detmar Meurers

RETUYT-INCO at BEA 2026 Shared Task 2: Meta-prompting in Rubric-based Scoring for German

Ignacio Sastre, Ignacio Remersaro, Facundo Díaz, Nicolás De Horta, Luis Chiruzzo, Aiala Rosá and Santiago Góngora

UOL@IDEM at BEA 2026 Shared Task 1: Neural Fusion and Feature-Rich Modeling for L1-Aware Vocabulary Difficulty Prediction

Nouran Khallaf and Serge Sharoff

15:30 - 16:00

Coffee Break

Saturday, July 4, 2026 (continued)

16:00 - 16:45 *Panel*

16:45 - 17:15 *Oral Session D*

Incentives Of EdTech: A Systematic Review Of EduNLP Research

Gabrielle Gaudeau, Aoife O’Driscoll, Jasper Degraeuwe, Andrew Caines, Donya Rooein and Zeerak Talat

Effects of Varying LLM Access on Essay Writing Behavior

Julia Christenson, Karin de Langis, Shirley Anugrah Hayati and Dongyeop Kang

17:15 - 17:30 *Closing Remarks*

Theory of Mind and Application in Educational Context

Effat Farhana¹, Maha Zainab¹, Qiaosi Wang², Niloofar Miresghallah²,
Ramira van der Meulen³, Max van Duijn³,

¹Auburn University, ²Carnegie Mellon University, ³Leiden University

{e.farhana, maz0032}@auburn.edu, qiaosiw@andrew.cmu.edu,

niloofar@cmu.edu, {r.van.der.meulen, m.j.van.duijn}@liacs.leidenuniv.nl

Abstract

This tutorial examines the integration of Theory of Mind (ToM) into AI-driven tutoring systems, with a focus on how large language models (LLMs) can represent learners' cognitive and emotional states to enable adaptive, personalized feedback. Participants will learn foundational ToM concepts from cognitive science and psychology and how these ideas can be operationalized in AI systems. We discuss mutual ToM, in which both tutors and learners model each other's mental states, and address challenges including misconception detection, metacognitive modeling, and privacy in data-driven tutoring. The tutorial also includes hands-on demonstrations of machine ToM in programming education using benchmark datasets such as CS1QA and CodeQA. By combining theoretical foundations, empirical insights, and practical exercises, this tutorial will provide an overview of designing human-centered, ethically aware, and cognitively informed AI tutoring systems.

1 Introduction

This tutorial explores the intersection of Theory of Mind (ToM) and AI-driven tutoring systems, focusing on how LLMs can be used to model students' cognitive states. We position this tutorial within the *cutting-edge* category and present a comprehensive overview of *evolution of cognitive modeling* in AI-driven tutoring systems: from rule-based and single-agent systems to reinforcement learning approaches and multi-agent LLM frameworks. Participants will learn about recent advances in this field, along with key limitations including privacy, mutual ToM, and the handling of student misconceptions, as well as practical tools and resources for deploying ready-to-use models in real-world educational NLP applications.

2 Tutorial Outline

The tutorial will be organized into three parts, covering the fundamentals of ToM, reflection, and ToM in educational contexts.

2.1 Part 1: Background (35 min)

Introduction (10 min) We will start with the core principles of ToM from cognitive science and psychology, explaining how humans attribute beliefs, desires, intentions, and emotions to themselves and others (Yeung et al., 2024). We will then discuss the importance of ToM and how ToM can be operationalized in human-centered AI systems (Mutaliyevich et al., 2025) to enhance human-AI interaction (Wang et al., 2024). Finally, we will motivate the integration of ToM into online tutoring systems, highlighting its potential to improve learner engagement, personalization, and adaptive feedback.

Modeling Cognition in AI Tutor (10 min) We will provide an overview of AI tutoring systems and the role of cognitive state modeling in such systems. We will discuss adaptive tutoring systems (MacLellan et al., 2016; MacLellan et al., 2014), emphasizing methods for real-time assessment of student knowledge (Feng et al., 2009), engagement (Baker et al., 2008), and learning difficulties. The discussion will illustrate how AI can provide personalized hints, guidance, and feedback based on learners' cognitive states (Abdelshiheed et al., 2024). We will also examine the challenges in modeling complex learning behaviors (Gao et al., 2023, 2024; Farhana et al., 2022) to give participants a clear understanding of the current capabilities and limitations of existing AI tutoring platforms.

LLM Adaptation in AI Tutoring (15 min) We will focus on the adaptation of LLMs in educational contexts: examples from programming, mathematics, and other domains where LLMs are used to provide explanations, hints, and interactive feedback. Examples include automated scoring of

math questions using LLM (Morris et al., 2025), automated feedback generation in argumentative writing (Baffour and Crossley, 2024), personalized coach in online learning (Dass et al., 2025). The discussion will highlight the strengths of LLMs in generating contextually appropriate responses while also pointing out limitations in recognizing and responding to students’ cognitive states. This sets the stage for exploring research gaps and the integration of ToM with LLM-based systems.

2.2 Part 2: Reflections and Questions (25 min)

In the first **15 minutes**, we will examine the gaps in current LLM-based tutoring systems and opportunities for ToM integration. We will also focus on the limitations of evaluating and measuring ToM in LLM-based systems (Wang et al., 2025; Chen et al., 2025; Ma et al., 2023; Ullman, 2023). Participants will understand implications of this gap and the need for ToM-informed AI in education. We will conclude the session with a **10-minute** reflection and open discussion outlining future research directions for the BEA community to advance human-centered evaluation methods and educational applications.

2.3 Part 3: ToM in Education (120 min)

We will cover core ideas of the tutorial in four interconnected themes: mutual ToM, addressing misconceptions, privacy considerations, and Machine ToM in source code comprehension.

Mutual ToM in AI Tutoring (25 min) We will introduce **mutual ToM**, where both the AI tutor and the learner maintain models of each other’s cognitive states (Wang and Goel, 2024; Ashktorab et al., 2025). This reciprocity allows the system to anticipate learner needs while learners understand AI intentions, improving trust, engagement, and learning outcomes. The discussion will cover strategies for implementing mutual ToM in online platforms, including cognitive state modeling, adaptive feedback, and interactive scenarios. Participants will explore how mutual understanding can be operationalized to enhance adaptive learning.

Addressing Misconceptions (25 min) This section examines the identification and correction of learners’ misconceptions in AI tutors. We will highlight the importance of integrating ToM principles to ensure that AI tutors do not overgeneralize or misinterpret learner behaviors. Recent research identifies common misconceptions in computational ToM, including the belief that AI should imple-

ment ToM as a singular module, that all social interactions require advanced ToM reasoning, that ToM is uniform across individuals, and that current LLMs already possess genuine ToM (van der Meulen et al., 2025). In practice, these insights suggest that LLM-based tutors should selectively apply cognitive modeling, account for variations in learners’ backgrounds and knowledge states, and avoid assuming ToM competence based solely on benchmark performance. Participants will learn methods for combining LLM pattern recognition with explicit cognitive modeling to detect misconceptions, deliver targeted feedback, and support students accurate reasoning. The section also highlights how misconception-aware tutoring can enhance learner engagement and outcomes while remaining aligned with human-centered AI principles.

ToM and User Privacy (25 min) Privacy and ToM are deeply intertwined: deciding what to share, with whom, and in what context requires reasoning about others’ knowledge, expectations, and interpretation of information. Education is a key domain where this is especially relevant, as learner data is sensitive and shared across multiple stakeholders, though the issue extends beyond education. We organize this section around a progression that mirrors how the field has come to understand the problem: from *whether* models reason about contextual privacy at all, to *what* they should disclose, to *how* users experience disclosures and inferences made on their behalf (Borkar et al., 2025).

We begin with the contextual integrity framing (Miresghallah et al., 2024a,b), which defines privacy as appropriate information flow rather than secrecy of fixed attributes. ConfAIde (Miresghallah et al., 2024b) shows that even frontier LLMs leak information in contexts where humans would not, and interventions such as privacy-inducing prompts and chain-of-thought reasoning fail to close the gap. The failure is linked to ToM, as models struggle in tracking knowledge and reasoning mental states of parties involved in an information exchange.

We then turn to persistent memory, which is increasingly central to LLM-based assistants. CIMemories (Miresghallah et al., 2025) shows that contextual integrity violations compound as memory accumulates and models are deployed across multiple tasks, with GPT-5 violations increasing from 0.1% to 9.6% across tasks and up to 25.1% under repeated prompts. The diagnosed failure mode is a *granularity failure* in which models identify the right information *domain* but cannot

distinguish relevant details from unnecessary disclosures. This naturally raises the next question of what a model *should* disclose, leading to data minimization as an operational principle (Zhou et al., 2025). Recent work shows LLMs are systematically biased toward oversharing and struggle to identify the minimal information they actually need—a capability gap, not just a policy gap.

Finally, we consider the user perspective, focusing on how LLMs not only *disclose* but also *infer* sensitive attributes such as (emotional states, health conditions, life circumstances) from interaction patterns. Recent user-centered (Monteiro et al., 2026) shows that users are more concerned with misrepresentative or unexpectedly shared inferences than with inference itself, and they prefer granular control over how such inferences are generated, stored, and shared. We conclude with best practices for privacy- and ToM-aware systems, particularly in education, emphasizing transparency of inferred states, user control over memory and inference scope, contextual-integrity-aware data flows, and default minimization in prompts/persistent storage.

Q&A and Machine ToM in Source Code Comprehension (45 min) We will present a demonstration of **Machine ToM**, including ToM among AI agents (Zhou et al., 2024) applied to programming and code comprehension tasks (Nikiema et al., 2025). The system models a learner’s misunderstanding patterns and selectively intervenes with explanations, or scaffolding (Bućinca et al., 2025). Participants will see how teacher–student LLM architectures simulate human-like tutoring interactions, guiding learners toward better reasoning and problem-solving (Liang et al., 2025). The demo will include examples of adaptive interventions and strategies for evaluating their effectiveness.

3 Specification of the Tutorial

Expected Audience Background The audience is expected to have foundational knowledge of NLP, machine learning, and evaluation methods, including supervised learning, classification, and metrics such as accuracy, precision, and recall. Familiarity with HCI, LLMs, and AI ethics is encouraged for participation in Q&A and discussions. We expect about 70–100 participants, based on growing interest in ToM and attendance at a recent workshop on [generative AI and ToM at IJCAI 2025](#) organized by one of the organizers.

Preferred Venue We strongly prefer ACL 2026 to

be held in the U.S. due to visa re-entry uncertainties, as many visa holders—including organizers, favor a U.S. location to reduce travel risks and support broad participation.

Diversity Considerations and Others This tutorial emphasizes in its instructional team, participant access, and content. The instructional team represents multiple career stages, from Ph.D. students to assistant professors, across four institutions on two continents and interdisciplinary backgrounds. To support accessibility, virtual participation will be available via Zoom. Approximately 33% of the session is devoted to related work from the wider research community, as reflected in the schedule and materials. The tutorial requires a standard Python environment with PyTorch. All code, data, and experimental scripts are openly released to facilitate reproducibility <https://github.com/MahaZainab/tom-in-education>.

Reading List To deepen participants’ understanding of ToM, we recommend the references cited in this tutorial as well as the readings listed below.

- SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents (Zhou et al., 2024).
- Large language models fail on trivial alterations to theory-of-mind tasks (Ullman, 2023).
- Towards Properly Implementing Theory of Mind in AI: An Account of Four Misconceptions (van der Meulen et al., 2025).
- Towards a Mutual Theory of Mind in Human-AI Interaction: How Language Reflects What Students Perceive About a Virtual Teaching Assistant (Wang, 2021).
- Framework for a multi-dimensional test of theory of mind for humans and ai systems (Stack et al., 2022).
- Can LLMs Keep a Secret? Testing Privacy Implications of Language Models via Contextual Integrity Theory (Miresghallah et al., 2024b).
- CIMemories: A Compositional Benchmark for Contextual Integrity of Persistent Memory in LLMs (Miresghallah et al., 2025).
- Operationalizing Data Minimization for Privacy-Preserving LLM Prompting (Zhou et al., 2025).
- When Are LLM Inferences Acceptable? User Reactions and Control Preferences for Inferred Personal Information (Monteiro et al., 2026).
- 1-2-3 Check: Enhancing Contextual Privacy in LLM via Multi-Agent Reasoning (Li et al., 2025).

- Mindless Dialogue? A Critical Note on ToM and Communicative Generative Agents (Farhana, 2025).
- Trust No Bot: Discovering Personal Disclosures in Human-LLM Conversations in the Wild (Mireshghallah et al., 2024a).

4 Presenter Biographies

Effat Farhana is Assistant Professor in the Department of Computer Science and Software Engineering at Auburn University. Her work has pioneered designing cognitive theory-informed AI systems for personalized learning, designing interpretable ML algorithms, and behavioral data mining within education and autism research. In 2021, she was named a Rising Star in Data Science by the University of Chicago for her PhD research. Effat is also an active community builder, serving as co-chair of the AI in Education Track at EAAI/AAAI 2024 and 2025, leading the organization of Theory of Mind workshop at IJCAI 2025, and serving as a Research Fellow at the NSF AI Institute for Adult Learning and Online Education (ALOE).

Maha Zainab is a Graduate Research Assistant in the Computer Science and Software Engineering Department at Auburn University. She is also a Gavin Graduate Student Fellow. She has participated in multiple international coding competitions and has also won some of them. Her research is in Generative AI, especially LLM evaluation, theory of mind and multi-agent reasoning. Her work is published in an IEEE conference, and her current research focuses on the Theory of Mind in intelligent systems. She has also taught national and international audiences.

Qiaosi Wang (Chelsea) is a Carnegie Bosch Postdoctoral Fellow at Carnegie Mellon University. Her work has proposed and empirically examined the Mutual Theory of Mind interaction paradigm to facilitate communications between online learners and AI agents in various social roles (e.g., teaching assistant, social facilitator) in online education. She has frequently presented her work at AI workshops and led the organization of the Theory of Mind in Human-AI Interaction (ToMinHAI) workshop series at multiple HCI venues (e.g., CHI, CUI).

Nilofar Mireshghallah is a Research Scientist at Meta AI's FAIR Alignment group in San Francisco. Beginning Fall 2026, she will join Carnegie Mellon University's Engineering & Public Policy (EPP) Department and Language Technologies Institute

(LTI) as an Assistant Professor. She received her Ph.D. from the CSE department of UC San Diego in 2023. Her research interests are privacy, natural language processing, and the societal implications of ML. She is a recipient of the National Center for Women & IT (NCWIT) Collegiate award in 2020 for her work on privacy-preserving inference, a finalist of the Qualcomm Innovation Fellowship in 2021, and a recipient of the 2022 Rising Star in Adversarial ML award.

Ramira van der Meulen is a Ph.D. student at Leiden University's Institute of Advanced Computer Science. She studies decision-making in Human-Machine Collaboration, with a special interest in its communicative requirements. Her current work focuses on the moment of 'agreement' in scenarios with incomplete information, where humans and AI establish their 'common ground' through varying levels of communication. This work concerns with Theory of Mind: the ability to take the perspective of others' beliefs, intentions and knowledge, and the use of this perspective to make sense of their behaviour and attitudes towards the world.

Max Johannes van Duijn is an Assistant Professor at Leiden University's Institute of Advanced Computer Science (LIACS) and Principal Investigator of the Social Intelligence Modelling [SIM] lab. He is a lecturer in Leiden's Data Science and Artificial Intelligence (DSAI) and Creative Science and Technology (CrIT) education programmes. In 2023 he was elected a member of The Young Academy, where he co-heads the Science and Society track. He is the associate editor for Machine Behaviour and Cognition of *Behaviour*. Research in the SIM lab combines methods from cognitive science, linguistics, and AI to study social intelligence, in particular empathy, perspective-taking, and Theory of Mind. Their work includes modeling these capacities in humans as well as in AI systems such as LLMs. His work is published in English and Dutch, in scholarly as well as popular venues, and has regularly attracted media attention.

5 Ethics Statement

As our tutorial will focus on modeling student cognition and providing personalized feedback, we will address ethical best practices for data collection and algorithmic design. We will emphasize inclusivity and fairness by supporting diverse learners and mitigating biases, aiming to advance safe, transparent, human-centered AI in education.

References

- Mark Abdelshiheed, Tiffany Barnes, and Min Chi. 2024. How and when: the impact of metacognitive knowledge instruction and motivation on transfer across intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 34(3):974–1007.
- Zahra Ashktorab, Djallel Bouneffouf, Krissy Brimijoin, Rachel Bellamy, Murray Campbell, Arielle Goldberg, Gabriel Enrique Gonzalez, Stephanie Houde, Miao Liu, Dario Andres Silva Moran, and 1 others. 2025. Bridging the gap: Unifying hci & ml perspectives on mutual theory of mind. In *International Joint Conference on Artificial Intelligence*.
- Perpetual Baffour and Scott Crossley. 2024. Advances in automating feedback for argumentative writing: Feedback prize as a case study. In *The Routledge international handbook of automated essay evaluation*, pages 303–328. Routledge.
- Ryan Baker, Jason Walonoski, Neil Heffernan, Ido Roll, Albert Corbett, and Kenneth Koedinger. 2008. Why students engage in “gaming the system” behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19(2):185–224.
- Jaydeep Borkar, Matthew Jagielski, Katherine Lee, Niloofar Mireshghallah, David A Smith, and Christopher A Choquette-Choo. 2025. Privacy ripple effects from adding or removing personal information in language model training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18703–18726.
- Zana Buginca, Siddharth Swaroop, Amanda E Paluch, Finale Doshi-Velez, and Krzysztof Z Gajos. 2025. Contrastive explanations that anticipate human misconceptions can improve human decision-making skills. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–25.
- Ruirui Chen, Weifeng Jiang, Chengwei Qin, and Cheston Tan. 2025. [Theory of mind in large language models: Assessment and enhancement](#). *arXiv preprint arXiv:2505.00026v1*.
- Rahul K Dass, Rochan H Madhusudhana, Erin C Deye, Shashank Verma, Timothy A Bydlon, Grace Brazil, and Ashok K Goel. 2025. Ivy: a hybrid knowledge-based and generative ai coach for explaining procedural skills. In *International Conference on Artificial Intelligence in Education*, pages 233–246. Springer.
- Effat Farhana. 2025. [Mindless dialogue? a critical note on theory of mind and communicative generative agents](#). In *Proceedings of the 12th Advances in Cognitive Systems*.
- Effat Farhana, Teomara Rutherford, and Collin F Lynch. 2022. Predictive student modelling in an online reading platform. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 12735–12743.
- Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19(3):243–266.
- Ge Gao, Xi Yang, and Min Chi. 2024. Get a head start: On-demand pedagogical policy selection in intelligent tutoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12136–12144.
- Qitong Gao, Ge Gao, Juncheng Dong, Vahid Tarokh, Min Chi, and Miroslav Pajic. 2023. Off-policy evaluation for human feedback. *Advances in Neural Information Processing Systems*, 36:9065–9091.
- Wenkai Li, Liwen Sun, Zhenxiang Guan, Xuhui Zhou, and Maarten Sap. 2025. [1-2-3 check: Enhancing contextual privacy in llm via multi-agent reasoning](#). *arXiv preprint arXiv:2508.07667*.
- Keyu Liang, Zhongxin Liu, Chao Liu, Zhiyuan Wan, David Lo, and Xiaohu Yang. 2025. [Zero-shot cross-domain code search without fine-tuning](#). *Proc. ACM Softw. Eng.*, 2(FSE).
- Ziqiao Ma, Jacob Sansom, Run Peng, and Joyce Chai. 2023. Towards a holistic landscape of situated theory of mind in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.
- Christopher J MacLellan, Erik Harpstead, Rony Patel, and Kenneth R Koedinger. 2016. The apprentice learner architecture: Closing the loop between learning theory and educational data. *International Educational Data Mining Society*.
- Christopher J MacLellan, Kenneth R Koedinger, and Noboru Matsuda. 2014. Authoring tutors with sim-student: An evaluation of efficiency and model quality. In *International conference on intelligent tutoring systems*, pages 551–560. Springer.
- Niloofar Mireshghallah, Maria Antoniak, Yash More, Yejin Choi, and Golnoosh Farnadi. 2024a. Trust no bot: Discovering personal disclosures in human-llm conversations in the wild. *arXiv preprint arXiv:2407.11438*.
- Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. 2024b. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *The Twelfth International Conference on Learning Representations (ICLR)*.
- Niloofar Mireshghallah, Neal Mangaokar, Narine Kokhlikyan, Arman Zharmagambetov, Manzil Zaheer, Saeed Mahloujifar, and Kamalika Chaudhuri. 2025. CIMemories: A compositional benchmark for contextual integrity of persistent memory in LLMs. *arXiv preprint arXiv:2511.14937*.

- Kyzyl Monteiro, Minjung Park, Alexander Ioffrida, Angelina Sanna, Niloofar Miresghallah, Yang Wang, and Sauvik Das. 2026. When are LLM inferences acceptable? user reactions and control preferences for inferred personal information. *arXiv preprint arXiv:2605.10013*.
- Wesley Morris, Langdon Holmes, Joon Suh Choi, and Scott Crossley. 2025. Automated scoring of constructed response items in math assessment using large language models. *International journal of artificial intelligence in education*, 35(2):559–586.
- Dedakhanov Abdumalik Mutalliyevich, S Balapriya, Deepak Dharrao, and Yuldashev Ulugbek Vokhidjon Ugli. 2025. Ai-based intelligent tutoring system for the cognitive development. In *AIP Conference Proceedings*, volume 3306, page 030072. AIP Publishing LLC.
- Serge Lionel Nikiema, Jordan Samhi, Abdoul Kader Kaboré, Jacques Klein, and Tegawendé F Bissyandé. 2025. The code barrier: What llms actually understand? *arXiv preprint arXiv:2504.10557*.
- Caoimhe Harrington Stack, Effat Farhana, Xinyu Shen, Simeng Zhao, Angela Maliakal, Roxanne Rashedi, Joel Michelson, and Maithilee Kunda. 2022. Framework for a multi-dimensional test of theory of mind for humans and ai systems. In *The Tenth Annual Conference on Advances in Cognitive Systems*.
- Tomer Ullman. 2023. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*.
- Ramira van der Meulen, Rineke Verbrugge, and Max J. van Duijn. 2025. [Towards properly implementing theory of mind in ai: An account of four misconceptions](#). *arXiv preprint arXiv:2503.16468*.
- Qiaosi Wang. 2021. [Towards mutual theory of mind in human-ai interaction: How language reflects what students perceive about a virtual teaching assistant](#).
- Qiaosi Wang and Ashok K. Goel. 2024. Mutual theory of mind for human-ai communication. In *Proceedings of the Workshop on Theory of Mind in Human-AI Interaction at CHI 2024*.
- Qiaosi Wang, Sarah Walsh, Mei Si, Jeffrey Kephart, Justin D Weisz, and Ashok K Goel. 2024. Theory of mind in human-ai interaction. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–6.
- Qiaosi Wang, Xuhui Zhou, Maarten Sap, Jodi Forlizzi, and Hong Shen. 2025. Rethinking theory of mind benchmarks for llms: Towards a user-centered perspective. In *Proceedings of the Human-centered Evaluation and Auditing of Language Models Workshop at CHI 2025 (HEAL @ CHI'25 Workshop)*, page 7, New York, NY, USA. ACM.
- Elaine Kit Ling Yeung, Ian A Apperly, and Rory T Devine. 2024. Measures of individual differences in adult theory of mind: A systematic review. *Neuroscience & Biobehavioral Reviews*, 157:105481.
- Jijie Zhou, Niloofar Miresghallah, and Tianshi Li. 2025. Operationalizing data minimization for privacy-preserving LLM prompting. *arXiv preprint arXiv:2510.03662*.
- Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haoifei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and 1 others. 2024. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*.

Inferring Student Engagement via Real-Time Thermal–Visual Voice Activity Detection

Bradley A. Goodman^{1,2}

¹The MITRE Corporation, ²Brandeis University,

Abstract

We introduce a thermal–visual fusion approach to improve non-invasive Voice Activity Detection (VAD) for classroom engagement monitoring. In noisy multi-speaker classrooms using a single microphone, acoustic-only methods fail to reliably isolate individual speakers. Our method integrates facial thermal signatures—capturing respiratory and speech-related heat patterns—with visual lip-motion cues to provide an acoustic-independent speech signal. This provides a localized, privacy-preserving, and acoustic-independent indicator of speech activity.

This system acts as a visual-diarization front-end, informing Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) systems not only when speech occurs, but precisely which student is speaking. Using up to 19 engineered features, our Thermal-Only Random Forest classifier achieved a Recall of 0.9234 and an F1-score of 0.8105 in subject-independent evaluations, outperforming visual-only baselines. The system was validated as a proof-of-concept on a Raspberry Pi 5 in a controlled laboratory setting, demonstrating real-time feasibility. These results demonstrate that thermal–visual fusion enables more reliable linguistic analysis of collaborative learning and provide critical input for AI agents to facilitate group participation in real-world educational settings that lead to more successful learning outcomes.

1 Introduction and Research Motivation

Voice Activity Detection (VAD) is fundamental to acoustic monitoring and classroom behavioral analysis. Existing solutions, outside of those focused

solely on audio, typically rely on visible-spectrum video and suffer from severe degradation in performance due to variability in lighting, occlusions, and person identity changes. The objective of this research was to investigate the efficacy of integrating facial thermal signature data—which captures the distinct heat patterns associated with respiration and articulation—to create a system that is robust and highly generalizable across different scenarios. Thermal cameras, once limited to industrial (Wilson et al., 2023) and medical (Ioannou et al. 2014; Manullang et al. 2021; Kwon et al. 2023) applications, have become increasingly viable for a wider range of uses due to recent reductions in both cost and size. The incorporation of Artificial Intelligence (AI) into classroom settings aims to promote effective peer learning by facilitating better team dynamics and ensuring equitable engagement among students. A critical component of this AI-Student teaming approach is the ability to accurately track active student engagement during collaborative tasks. Previous research into engagement detection in noisy classroom environments has faced significant challenges:

- **Acoustic Limitations:** Efforts relying on speech understanding and speaker identification were hindered by the use of a single microphone and the highly noisy nature of multiple peer groups working concurrently in a classroom. This resulted in relatively poor speech recognition with high Word Error Rates (WER) and an inability to reliably isolate the speaker (Cao et al. 2023; Wang et al. 2024). These acoustic failures often prevent NLP systems from performing accurate group dynamics analysis, knowledge-sharing tracking, or peer-learning modeling, as the input text is either missing or heavily corrupted.

- **Visual Limitations:** Multi-modal gesture recognition (e.g., pointing or looking at a peer) successfully determines engagement but fails to account for instances when a student is actively

¹Approved for Public Release; Distribution Unlimited. Public Release Case Number 26-0344. The author’s affiliation with The MITRE Corporation is provided for identification purposes only and is not intended to convey or imply MITRE’s concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author.

speaking but not looking directly at another student or activity-related object (Palmer et al., 2025). Furthermore, attempts at purely visual lip-reading are highly restrictive, requiring the speaker’s mouth to be clearly aligned with the camera, which is often impractical in a dynamic classroom. Lip-reading can, however, potentially increase the accuracy of speech recognition (Yoshida et al. 2008; Guan et al. 2024; Guan et al. 2025). Given the inherent limitations of both purely acoustic and constrained visual methods, a more passive and robust detection mechanism is required to reliably determine who is speaking, regardless of noise or precise facial orientation.

2 Related Work

Foundations of Collaboration and Engagement:

Early efforts in modeling team dynamics by Goodman et al. (2006) established that individual and group user models could detect faltering collaboration. By utilizing indicators such as dialogue acts, backchannel utterances (e.g., "uh-huh"), and prosodic speech features, they demonstrated that machine learning (ML) can effectively predict listener engagement and shared focus in distributed environments. Similarly, Oztoprak (2022) confirmed that prosodic cues alone—specifically pitch and intensity—are significant predictors of emotional engagement in child-robot interactions, achieving 73.9% accuracy with artificial neural networks. D’Mello et al. (2012) developed a model to explain the dynamics of affective states during learning activities that included states of engagement. Their model establishes engagement as a dynamic, temporally evolving cognitive–affective process, motivating the need for sensing approaches that can capture rapid state transitions during authentic learning activities. While the early research focused on prosodic features and affective states (Cao et al., 2014), more recent work has explored automated scoring of collaborative classroom discussion (Tran et al., 2025).

Audio-Visual and Nonverbal Modalities:

Building on these theoretical foundations, subsequent work has operationalized engagement through observable behavioral and physiological signals such as facial expression, gaze, head motion, and speech. Bosch et al. (2016) showed that visible-spectrum video features, including facial expressions and head pose, can infer learner affect in classroom settings, while more recent multi-

modal approaches (e.g., Sharma et al., 2023) incorporate eye tracking and head movement to improve engagement detection. However, these methods remain sensitive to lighting conditions, occlusion, and expressive variability, limiting robustness in unconstrained educational environments. To address these limitations, researchers have explored multimodal architectures. Tao et al. (2021) proposed TalkNet, an active speaker detection framework that uses audio-visual cross-attention and long-term temporal features to align lip motion with speech, improving performance under overlapping speech and visual noise. It processes raw video instead of running in real time. To empirically assess TalkNet’s applicability as a baseline for thermal-domain VAD, we ran TalkNet on the same thermal false-color video used for real-time evaluation. TalkNet detected a stable face track across 3,218 frames (128.7 seconds), yet only 54 frames (1.7%) were classified as speaking, compared to 72% ground-truth speech in the same session. The audio-visual synchrony model relies on visible-spectrum lip texture to align mouth motion with audio; thermal false-color imagery suppresses this texture, rendering the model’s visual stream uninformative regardless of acoustic content. This confirms that RGB-trained audio-visual models do not transfer to thermal input without retraining, empirically motivating the purpose-built thermal feature pipeline described in Section 3. Abdrakhmanova et al. (2021) introduced the large SpeakingFaces dataset, combining thermal, visual, and audio recordings of voice commands across 142 subjects; however, its clips range from 2–7 seconds (50–175 frames at 28 fps), each beginning with a brief silence, which is insufficient to populate the 125-frame rolling window required for our spectral breathing features (*Breathing_Power_In_Band_RA*, *Breath_BRSI*) to stabilize — causing both features to evaluate to zero across the entire dataset and producing near-random classification when the trained model was applied to continuous thermal video. Palmer et al. (2025) further argue that speech recognition alone is insufficient in noisy classrooms and advocate integrating nonverbal cues such as gaze, posture, and gesture for context-aware modeling of group interaction and engagement.

Classroom Implementation and Constraints:

Translating these models to the classroom remains a primary challenge. Xu et al. (2023) successfully predicted tutoring engagement using a combination of face exposure and speech duration met-

rics. In more complex group settings, Wang et al. (2024) utilized Whisper ASR and SpeechBrain for speaker diarization, achieving a moderate correlation (≈ 0.62) with human annotations despite the high-noise classroom environment. Baker et al. (2014) examined cost-effective engagement detection.

Research Gap: While these studies leverage prosodic, acoustic, and visual cues, they remain susceptible to background noise, ambient lighting variability, and the physical occlusions or motion artifacts inherent in unconstrained settings. Our research addresses this gap by proposing a passive, multi-modal fusion of thermal and visual signatures, offering an acoustic-independent physiological signal for reliable VAD in real-time.

3 Research Hypothesis and Methodology

Our core hypothesis is that thermal imaging of the mouth and nasal regions can provide a novel, reliable, and passive method to determine speech activity by tracking the localized temperature changes caused by respiration. We further hypothesize that a fusion model integrating this thermal data with robust lip motion data will provide a superior capability to detect speech even when the speaker is not fully facing the camera (e.g., Bennett et al. (2020) showing side-view of respiratory plume using a thermal camera).

To test this hypothesis, we collected thermal videos of individuals engaging in both episodes of speaking and non-speaking in a laboratory setting. The thermal video set included single and multi-party participants (Figure 1). Using the trained Yolov8n face-detection model (Vemulapalli et al., 2023), we isolated the face, mouth and nasal regions in each thermal video frame to permit localized extraction of temperatures. This visual tracking data was combined with automatic audio annotation using a widely-used neural-acoustic VAD tool (Silero VAD (Silero, 2024)) to segment periods of active speech. From this multi-modal data stream, we developed a suite of metrics—including rolling thermal averages, derived thermal deltas, and continuous lip motion kinetics—which were then used to train and evaluate ML classifiers for speech prediction.

Our research focuses on one aspect of student engagement detection, episode-level speech detection, which can be fused with other forms of engagement such as gesture, pointing, gaze, and speech recog-



Figure 1: Single and multiple participant data collection.

nition.

3.1 Dataset and Preprocessing

The final dataset used for model development and evaluation included synchronized audio, thermal video, and their associated feature streams. The raw ground truth was generated using the open-source Silero Voice Activity Detector (VAD) that looked for speech activity versus background noise, labeling each frame with a binary Speech_Present flag (TRUE/FALSE) based on whether or not the probability of speaking exceeded a certain value. As the ground truth source, Silero VAD achieves perfect agreement with its own labels by definition; the thermal classifier’s performance reported in Section 4 therefore represents the degree to which thermal physiological signals can replicate an acoustic oracle operating under controlled laboratory conditions. A total of 40,855 frames (approximately 27.2 minutes at 25 FPS) were aggregated from twenty distinct recording sessions, encompassing both single- and multiple-subjects. The single-subject data (17 distinct recording sessions, $\approx 28,000$ frames) formed the foundation for our Generalization Test Set. For rigorous model evaluation, the entire dataset was partitioned using Leave-One-Experiment-Out (LOEO) cross-validation. This methodology ensured strictly subject- and experiment-independent testing by training the model on the data from 16 experiments and validating its performance on the single remaining, unseen experiment. Six model types were considered during LOEO: Fusion Random Forest (RF) (Liaw and Wiener, 2002), Visual-Only RF, Thermal-Only RF, Fusion Multi-Layer Perceptron (MLP) (Chan et al., 2023), Fusion Support Vector Machine (SVM) (Noble, 2006), and Histogram-based Gradient Boosting Classification (HGBC) (Guryanov, 2019).

Frame-level VAD is inherently susceptible to noise, resulting in intermittent suppression of Speech_Present flags during continuous speech segments due to brief drops in vocal energy dur-

ing breath inhalation or soft phoneme articulation. Rather than applying a rolling mean to the binary signal, we addressed this at the source by setting Silero VAD’s internal confidence threshold to 0.70, which suppresses low-confidence transient detections while preserving sustained speech intervals. Each detected speech interval was then extended forward by 0.3 seconds (1–2 frames at 5 fps) to capture residual thermal signal that persists briefly after speech cessation. To compensate for an initial estimate of the audio-thermal synchronization lag, labels were shifted 2 frames earlier prior to assignment; subsequent manual alignment revealed the true lag to be 12 frames, as discussed in Section 4.2. Finally, during training, segments shorter than 8 frames (1.6 seconds) were excluded to remove transition-zone frames where the lag correction is insufficient to guarantee correct labeling. Before this segment filter, approximately 77% of frames carried a TRUE label; after filtering, the proportion remained stable at 76–78% across sessions, confirming that the filter removed short segments of both classes proportionally rather than biasing the label distribution.

- **Thermal Data:** Extracted features from thermal camera at 25 frames per second (AMPBANK M256 thermal camera, 256x196 pixels, 56° x 42° Field of View) focused on the thermal gradient changes around the mouth-nasal regions, indicating exhalation and subtle muscle movements during speech and sent to a local Raspberry Pi 5 8GB.
- **Audio Data:** Audio from a single USB microphone.
- **Preprocessing:** All data streams were normalized and aligned temporally before being passed to the ML architecture. The YOLOv8 model (Vemulapalli et al., 2023) was used for face, nose, and mouth detection in each thermal video frame as the frames were assembled and aligned with the audio stream. Thermal data was extracted and stored into a CSV file.

3.2 Feature Engineering and Preprocessing

The raw thermal and visual data were rigorously preprocessed to derive the final feature set used in the analysis. Crucially, due to the high volatility of instantaneous thermal readings, we introduced Temporal Rolling Average (RA) features to capture the sustained thermal signature of speech activity. Table 1 lists the thermal and visual features collected and calculated. These features do not just detect activity; they provide the temporal ‘gating’

required to provide corroborating evidence of ASR events when a specific student is speaking, thereby reducing ‘hallucinations’ or noise-injection.

3.3 Model Architecture and Optimized Validation

We evaluated the fusion approach using a refined dataset of 17 single-speaker subject-sessions (28,000 frames) to establish a reliable generalization baseline. A rigorous Leave-One-Experiment-Out (LOEO) cross-validation was employed. After systematic hyperparameter tuning of candidate architectures—including Histogram-based Gradient Boosting Classification (HGBC) and Multi-Layer Perceptron (MLP)—the Random Forest (RF) classifier emerged as the optimal architecture. HGBC is a ML algorithm designed to train efficiently on large datasets. It speeds up the training process by grouping continuous feature values into discrete bins, which simplifies how decision trees determine split points (Guryanov, 2019). To maximize temporal stability, a 125-frame Rolling Average (RA) window was applied to the final 19-feature set, as a temporal sweep confirmed this window size provided the highest subject-independent generalization.

4 Results and Performance Analysis

The comparative classifier evaluation study (Table 2) confirms that thermal-only and thermal-visual fusion significantly outperform the visual-only baseline.

4.1 Discussion of Performance Metrics

The Thermal-Only RF (no *MAR_Variance*) achieved F1=0.8105 and Recall=0.9234 under rigorous LOEO cross-validation, confirming the system captures physiological markers of respiration and articulation rather than superficial environmental cues. The 11.64% F1 gain over the visual baseline (Visual-Only_RF) validates that thermal features provide non-redundant complementary information. While Precision (0.7308) indicates some false alarms from heavy-breathing artifacts, the high Recall confirms thermal signatures are the primary drivers for capturing continuous speech indicators. The Fusion_RF model ranked second, with *MAR_Variance* providing the greatest visual contribution.

Feature	Type	Signal captured	Speech vs. quiet
<i>Optimised temporal thermal cues (rolling averages)</i>			
Avg_Mouth_Temp_RA ₁₂₅	Thermal	Mean mouth temp, 125-frame rolling avg	Elevated during speech
Avg_Nostril_Temp_RA ₁₂₅	Thermal	Mean nostril temp, 125-frame rolling avg	Lower during speech
Center_Mouth_Temp_RA ₁₂₅	Thermal	Center-mouth temp, 125-frame rolling avg	Elevated during speech
<i>Engineered thermal contrast</i>			
Delta_AvgTemp_Mouth_Roll_Avg	Thermal	Rolling mean face—mouth temperature delta	Increases during speech
Breathing_Power_In_Band_RA	Spectral	Nostril spectral power, 0.1–0.5 Hz band	Lower during speech
Breath_BRSI	Spectral	Nostril rhythm stability (100-frame FFT)	Lower during speech
Delta_Mouth_Min	Thermal	Mean mouth temp minus min face temp	Higher during speech
Center_Mouth_Contrast	Thermal	Center-mouth / mean-mouth temperature	Higher during speech
Center_Mouth_Delta_Face	Thermal	Center-mouth minus mean face temperature	Higher during speech
<i>Instantaneous thermal components</i>			
Avg_Mouth_Temp_C	Thermal	Mean mouth ROI temperature (current frame)	Elevated during speech
Avg_Nostril_Temp_C	Thermal	Mean nostril ROI temperature	Lower during speech
Avg_Temp_C	Thermal	Mean temperature across full face ROI	Baseline reference
Center_Mouth_Temp_C	Thermal	Center-mouth temperature (current frame)	Elevated during speech
Min_Temp_C	Thermal	Minimum temperature across face ROI	Lower during speech
Nostril_Temp_Diff_Min	Thermal	Mean nostril temp minus min face temp	Higher during speech
Face_Temp_Spread	Thermal	Max—min temperature across face ROI	Higher during speech
Nostril_Temp_Max_Slope	Thermal	Max $ dT/dt $ nostril, 100-frame window	Lower during speech
Mouth_Temp_Max_Slope	Thermal	Max $ dT/dt $ mouth, 100-frame window	Higher at onset
<i>Visual mouth motion</i>			
MAR_Variance	Visual	Mouth aspect ratio variance, 25-frame	Higher during speech

Revised feature set: A subsequent 13-feature revision replaces the three RA₁₂₅ features and redundant instantaneous temperature columns with slope-based dynamics (*Airflow_Dominance*, *Slope_Coupling*, *Mouth_Slope_Accel*) and spectral coupling features (*NM_Xcorr*, *Mouth_Breath_BRSI*). A 12-feature thermal-only version drops *MAR_Variance*. Results using the revised set are reported in Table 2 and discussed in Section 4.3.

Table 1: Thermal and visual features used in the original 19-feature classifier. $S_m = \text{Max } |dT/dt|$ mouth, $S_n = \text{Max } |dT/dt|$ nostril: max temperature slopes in mouth and nostril ROIs. RA₁₂₅ denotes a 125-frame rolling average.

4.2 Feature Importance Analysis

For the original 19-feature model (Table 1), the Permutation Feature Importance rankings (Table 3) provide a clear "biological" map of how the model makes these decisions. Feature importance analysis substantiates the top thermal cues that drive the model’s predictive power. The top feature, *Center_Mouth_Delta_Face*, highlights that the temperature difference between the mouth and the rest of the face is the most critical indicator of speech. This is closely followed by *Breathing_Power_In_Band_RA* which is calculated over a continuous rolling window of 500-frame and then a 125-frame window for smoothing, which confirms that rhythmic respiratory signals captured by the thermal sensor are vital for distinguishing speech from silence. Notably, the many negative values at the bottom of the list (such as *Mouth_Temp_Max_Slope*) reveal multicollinearity: because many thermal features are highly correlated, the model sees them as redundant. These results suggest the model might be simplified by focusing only on the top 5–6 unique thermal and breathing markers without losing sig-

nificant performance. Correlation analysis reveals that the feature set naturally separates into 3 partially independent clusters corresponding to baseline facial thermal state, thermal gradients associated with respiratory airflow, and breathing rhythm and motion dynamics. This separation suggests that the classifier exploits complementary physiological signals rather than redundant measurements.

The visual cue, *MAR_Variance*, ranked first overall in a Permutation Importance Analysis of the Fusion RF Model (results not shown here). *MAR_Variance* captures continuous lip deformation changes rather than binary motion events. Together, the Permutation findings for the Thermal-Only RF and Fusion RF models demonstrate that both model’s performance gains arise from carefully engineered feature sets that utilize stable thermal physiology or refined visual motion descriptors, leveraging the strengths of both modalities.

Manual alignment of VAD labels against thermal frames revealed a systematic 12-frame lag (2.4 s at 5 fps) from Silero VAD onset latency combined with audio-thermal synchronization offset. Correcting this lag in a revised 12-feature thermal-only

Model	F1	Accuracy	Precision	Recall	Δ F1	Training Details
<i>Visual-Only_RF</i>	0.7260	0.6196	0.7413	0.7150	–	500 trees (tuned)
<i>Thermal-Only_RF</i> ^a	0.8105	0.7079	0.7308	0.9234	+11.64%	500 trees (tuned)
<i>Thermal-Only_RF(revised)</i> ^b	0.8300	0.7407	0.7830	0.8967	+14.33%	500 trees (tuned)
<i>Fusion_MLP</i>	0.6574	0.5878	0.7240	0.6286	–9.45%	layer_sizes=(100,50,25), max_iter=500
<i>Fusion_HGBC_Opt</i>	0.6921	0.6455	0.7475	0.7011	–4.67%	100 epochs
<i>Fusion_SVM</i>	0.7048	0.6369	0.7204	0.7385	–2.92%	optimized-based
<i>Fusion_RF</i>	0.8047	0.7000	0.7270	0.9183	+10.84%	500 trees (tuned)

^a Silero VAD threshold=0.70, 0.3s hysteresis tail, 2-frame onset correction, 8-frame transition-boundary exclusion (76.2% TRUE frames). Without segment filter: F1=0.8133, Accuracy=0.7192, Precision=0.7734, Recall=0.8693.

^b Revised 12-feature thermal-only set, 18 sessions including one balanced session (37% speech); *NM_Xcorr* and *Mouth_Breath_BRSI* rise to top-2 permutation importance after 12-frame VAD lag correction (see Section 4.2); adding *MAR_Variance* gives the 13-feature Fusion RF variant (F1 = 0.8267).

Table 2: Model performance comparison.

model elevated *NM_Xcorr* (nostril–mouth temperature correlation, which decouples during speech as oral exhalation displaces nasal breathing) to rank 1 (+0.030) and *Mouth_Breath_BRSI* (oral breathing rhythm stability, disrupted by phonation) to rank 2 (+0.028), with 11 of 12 features positive; the sole exception, *Nostril_Temp_Max_Slope* (–0.005), is subsumed by *NM_Xcorr* and *Airflow_Dominance* which capture nostril-mouth dynamics more directly. Both features had shown negative importance in prior runs attributed to insufficient quiet data; the lag correction reveals systematic label misalignment as the true cause, validating thermal channel decoupling as the dominant physiological marker of speech. Revised results are shown in Table 2 and discussed in Section 4.3.

4.3 Future Work and Generalization

Iterative Feature Refinement. Correlation analysis of the original 19-feature set (Table 1) revealed substantial redundancy among the three 125-frame rolling average temperature features and the instantaneous temperature columns from which they are derived, as these cluster into a single highly-correlated group encoding overall facial thermal state rather than speech-specific dynamics. A revised 12-feature thermal-only set replaces these redundant columns with slope-based dynamics (*Airflow_Dominance*, *Slope_Coupling*, *Mouth_Slope_Accel*) and spectral coupling features (*NM_Xcorr*, *Mouth_Breath_BRSI*) that more directly characterize the physiological contrast between oral speech exhalation and quiet nasal breathing. Evaluated on 18 sessions—including one session with extended alternating quiet and speech periods (37% speech, compared to a mean of 77% across the original 17 sessions)—and with

an 8-frame minimum segment filter and 12-frame VAD label correction applied at training time, the revised Thermal-Only RF achieved F1 = 0.8300, Accuracy = 0.7407, Precision = 0.7830, and Recall = 0.8967 (Table 2). The 12-frame offset (2.4 s at 5 fps) reflects the combined effect of Silero VAD’s onset latency and audio-thermal synchronisation delay, as identified through manual annotation alignment and discussed in Section 4.2. Transitioning this approach into a high-confidence classroom solution requires refinement in three key areas.

4.3.1 Feature and Tracking Robustness

The success of the temperature differences and rhythmic respiratory signals confirms that temporal context is paramount. Future iterations will explore advanced time-series features to better characterize the frequency of speech expulsion. Furthermore, replacing the current 5-point YOLOv8n landmarks with high-density facial models (e.g., 68-point landmarks) could increase the predictive weight of visual features like *MAR_Variance*. Finally, to handle multi-student interactions, we intend to replace our basic tracker with Deep SORT (Wojke et al., 2017) to maintain subject identity across thermal transients and occlusions. An initial effort with Deep SORT maintains identity through head turns via joint appearance embedding and Kalman-filter position prediction, holding tracks alive across brief occlusions rather than creating new identifiers when a face temporarily exits the detector.

4.3.2 Dataset Expansion and Kinematics

By expanding the dataset to include longer data collection sessions, a broader population, and collecting data across various speech styles and ambi-

Rank	Feature	Avg.	Rank	Feature	Avg.
1	<i>Center_Mouth_Delta_Face</i>	0.0031	10	<i>Center_Mouth_Temp_C</i>	-0.0014
2	<i>Breathing_Power_In_Band_RA</i>	0.0030	11	<i>Delta_AvgTemp_Mouth_Roll_Avg</i>	-0.0024
3	<i>Nostril_Temp_Diff_Min</i>	0.0018	12	<i>Avg_Temp_C</i>	-0.0026
4	<i>Avg_Nostril_Temp_C</i>	0.0017	13	<i>Min_Temp_C</i>	-0.0027
5	<i>Delta_Mouth_Min</i>	0.0012	14	<i>Avg_Mouth_Temp_C</i>	-0.0035
6	<i>Breath_BRSI</i>	0.0004	15	<i>Center_Mouth_Temp_125Frame_RA</i>	-0.0038
7	<i>Center_Mouth_Contrast</i>	-0.0006	16	<i>Avg_Nostril_Temp_125Frame_RA</i>	-0.0041
8	<i>Nostril_Temp_Max_Slope</i>	-0.0011	17	<i>Avg_Mouth_Temp_C_125Frame_RA</i>	-0.0053
9	<i>Face_Temp_Spread</i>	-0.0012	18	<i>Mouth_Temp_Max_Slope</i>	-0.0067

Table 3: Permutation feature importance analysis (Thermal RF model). *MAR_Variance* (visual) excluded from Thermal-Only RF; 18 remaining features shown.

ent temperatures, the model will be forced to learn the universal physical principles of speech kinematics and thermal signatures. This is essential for achieving a high-confidence, deployable solution that generalizes across multiple individuals and conditions.

4.3.3 Integrated Multimodal Framework for Classroom Dynamics

Our long-term goal is to transition from a supplementary thermal VAD to a fully integrated framework for analyzing peer-learning dynamics such as Palmer et al. (2025). Thermal signatures will be fused with gaze, posture, and gesture recognition (VanderHoeven et al., 2023) to triangulate referential intent, and coupled with lip-reading (Guan et al., 2025) and diarization (Wang et al., 2025) to improve acoustic resilience in high-noise classrooms. This combined approach should achieve better results at speech and engagement detection.

5 Towards a Real-Time Implementation and Deployment – Engagement Detection System

5.1 Edge-Computing Architecture

The objective of this real-time system is to assess whether a thermal-visual VAD can reliably identify one form of student engagement, defined here as the presence of speaking activity overlapping with true speaking episodes. The system is not intended to perform precise speech segmentation; instead, it aims to robustly detect whether a student speaks during an interaction which can indicate active engagement in a peer-learning activity. The evaluation of the system focuses on episode-level detection and engagement-oriented metrics rather than frame-level accuracy that was pursued during classifier training.

To evaluate the practical utility of the thermal-

visual fusion model, we developed a custom Python runtime on a local and cost-effective Raspberry Pi 5. The system utilizes an AMPBANK M256 thermal camera. Facial landmarks are isolated using a YuNet/YOLOv8-face algorithm, while thermal statistics from the mouth and nasal regions are extracted and summarized over the 125-frame rolling window.

5.2 Adaptive Calibration and Hysteresis

To improve classification robustness across individuals and recording environments—and to account for deployment conditions that may differ from those encountered during classifier training—we introduced a silent buffering period followed by an interactive calibration phase at the beginning of each session. The buffering period, during which participants remained silent, lasted 20 or more seconds (depending on the frame rate achieved by the Raspberry Pi 5 processor) and served to stabilize the thermal camera and determine participants’ breathing rates while allowing participants to acclimate to the data collection environment. During the subsequent calibration phase, participants first remained silent for 5 seconds and then spoke continuously for 5 seconds. Median classifier prediction probabilities from these two segments were used to compute an adaptive decision threshold T_{base} .

This buffering and adaptive calibration strategy successfully stabilized speaking-state detection for a single speaker across short (one to three minute) test sessions under varying ambient conditions. However, thermal speech detection remains sensitive to head and upper-body motion: movement can displace the air surrounding the face, mouth, and nostrils, causing thermal signatures to lag behind physical landmarks and resulting in transient temperature artifacts. Such effects can lead to false-positive speech detections.

To address this, the speech detection pipeline fol-

lows a staged decision process designed to enforce temporal stability under noisy thermal conditions. First, the Random Forest classifier produces a raw frame-level speech probability. This signal is temporally smoothed using a rolling median filter to suppress short-lived fluctuations. Next, a motion-gating mechanism (Meyers et al., 2022) computes the Euclidean distance between the tracked face bounding box centers in consecutive frames. If the displacement exceeds a predefined threshold, a Motion Lock is triggered, and the current thermal contribution is suppressed by substituting a neutral probability value. This prevents thermal transients caused by head movement from influencing the speech decision.

The stabilized probability is then passed through a hysteresis-based state machine, a standard control strategy for preventing rapid state oscillations (Meister et al., 2015), with separate entry and exit thresholds derived from T_{base} . This mechanism prevents rapid state oscillations near the decision boundary—analogueous to thermostat control—by requiring stronger evidence to enter the speaking state than to remain in it.

5.3 Engagement Confidence Index

Because the objective of the classroom system is to identify active peer interaction rather than precise speech segmentation, we developed the Engagement Confidence Index (ECI). ECI is a composite score that estimates the likelihood and strength of a user’s future engagement based on observed behavioral signals and interaction patterns. This metric prioritizes episode-level detection over frame-level accuracy. ECI and a subset of related metrics we used are defined in Table 4.

5.4 Runtime Results and Discussion

Only a small initial test has been conducted using three different subjects and five thermal video sessions. Figure 2 shows probability trajectories and adaptive decision boundaries during runtime for a single subject. For the evaluated session, Engagement Recall was 0.620, Engagement Precision was 0.795, temporal stability score was 0.826, Episode Recall was 0.400, and ECI was 0.551. The gap between frame-level Recall (0.923 under LOEO cross-validation; 0.788 in the Figure 2 runtime session) and Episode Recall (0.400) reflects that short episodes fall below the temporal smoothing threshold or are interrupted by Motion Lock; the conservative calibration strategy favors precision over

Metric	Definition
Engagement Confidence Index (ECI)	Multi-objective engagement score over a temporal window; clamped to $[0, 1]$ with weights $w_1 = 0.5, w_2 = 0.3, w_3 = 0.2$: $\text{clamp} \left(\frac{w_1 R_{\text{ep}} + w_2 C_{\text{speech}}}{+ w_3 S_{\text{temp}}}, 0, 1 \right)$
Episode Recall (R_{ep})	Fraction of GT speech episodes matched by a predicted ENGAGED episode (IoU ≥ 0.5 , one-to-one, ± 1 frame tolerance): $R_{\text{ep}} = N_{\text{matched}} / N_{\text{GT}}$
Coverage (C_{speech})	Temporal overlap between predicted engagement and GT speech: $\sum_i (\text{Engaged}_i \cdot \text{GT}_i) / \sum_i \text{GT}_i$
Temporal Stability (S_{temp})	Penalty for rapid state flicker where $F = \text{Transitions} / \text{Duration}_{\text{sec}}$ $e^{-0.5F}$

Table 4: Engagement Confidence Index (ECI) and component metrics.

episode-onset sensitivity. These results suggest the acoustic-independent thermal stream, stabilized by temporal smoothing, provides a useful engagement signal in real time.

The Episode Recall of 0.400 warrants interpretation in the context of the deployment pipeline. A speech episode is counted as detected only if a predicted ENGAGED interval overlaps the ground-truth episode by at least 50% (IoU ≥ 0.5 , where IoU is the ratio of the temporal intersection to the union of predicted and ground-truth episode boundaries). Three pipeline stages each independently reduce the probability of achieving this threshold. First, the 125-frame rolling window introduces an onset delay before features stabilize at the start of a new speaking episode, causing the classifier to respond slowly at speech boundaries. Second, the Motion Lock mechanism suppresses thermal contributions during head movement, which frequently co-occurs with speech onset as a speaker turns to address a peer. Third, the conservative hysteresis entry threshold—set to favor precision over recall to minimize false-positive alerts to the AI agent—requires sustained evidence before transitioning to the ENGAGED state, penalizing short episodes. Together these stages create a system that detects the interior of long speaking episodes reliably (reflected in the 0.923 frame-level Recall) while frequently missing the onset of short or movement-accompanied episodes (reflected in the 0.400 Episode Recall). For the classroom engagement use case this tradeoff is acceptable: the system is designed to confirm sustained engagement rather than react to every utterance, and false-

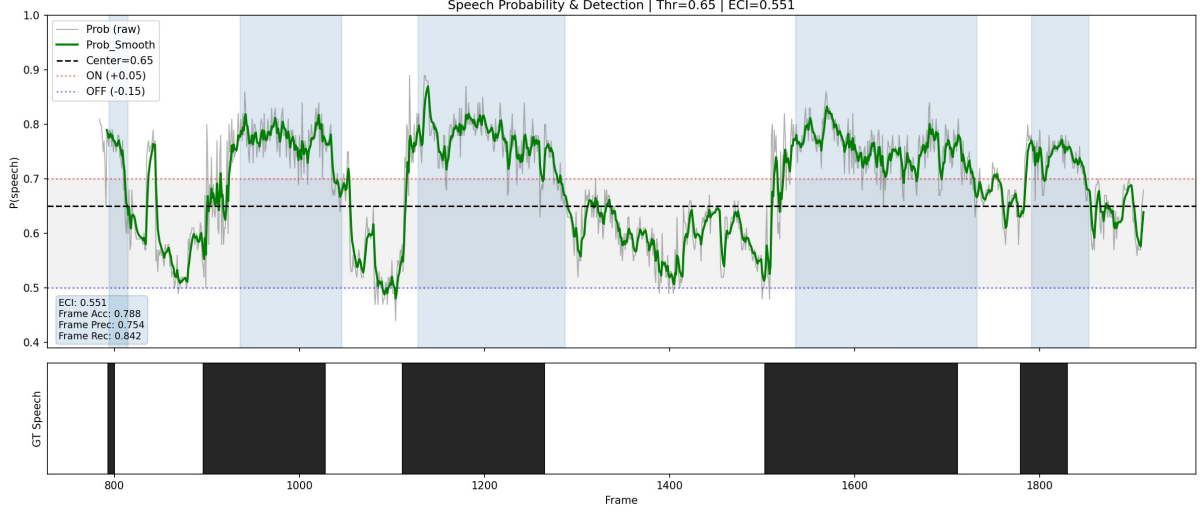


Figure 2: Real-time speech probability trajectories for a single subject. Gray: raw RF output; green: 5-frame smoothed probability; black dashed: calibrated threshold T_{base} ; red/blue dotted: hysteresis ON/OFF boundaries; light blue shading: predicted speaking intervals. Ground truth shown below (black = speaking, white = quiet).

positive interventions by an AI agent are more disruptive than missed short episodes.

Figure 3 summarizes the overall system architecture, highlighting the interaction between offline model training and the real-time inference pipeline incorporating buffering, adaptive calibration, motion gating, and hysteresis.

6 Conclusion and Future Impact

This research moves towards a high-confidence, acoustic-independent solution for detecting student speech in noisy, multi-peer environments. By successfully validating the fusion concept on a Raspberry Pi 5, we demonstrate the feasibility of an edge-computing architecture for classroom deployment. Future work will transition this proof-of-concept into a split-computing framework where the Raspberry Pi performs real-time feature extraction while a remote pipeline (Zhu et al., 2026) on an HPC platform fuses multi-modal components—gaze, posture, gesture, speaker diarization, and spoken language understanding—to build educational applications. The future work will integrate the thermal-visual front-end with large language models (LLMs) to provide real-time feedback on group dynamics, enabling AI agents to facilitate more equitable and linguistically-rich peer learning interactions.

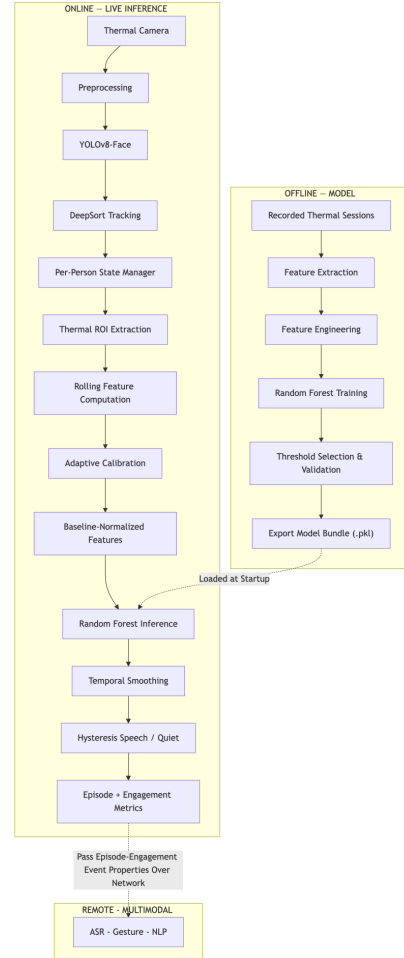


Figure 3: Online live inference (real-time) engagement detection system, remote multimodal processing, and offline model training system.

Limitations

The current focus of this research was modeling each person’s thermal properties in a classroom peer group to determine whether or not someone was speaking and whether or not they were engaged in the learning activity. Some limitations exist in the current implementation:

- Thermal camera resolution and field-of-view become an issue as the scene includes more people. The ROIs will cover fewer pixels which reduces the quality of thermal data. We must explore how this affects our models if we are to continue to utilize inexpensive thermal cameras.
- A simplistic face tracker was employed but it was not effective in continuously distinguishing each person as the same person especially when one turned his or her head quickly or moved. For that reason, the current research results focused on one person to better model thermal characteristics of speaking or not speaking. Future work must examine multiple participants in a scene since that is what one expects in a classroom peer-learning event. Better face tracker algorithms, such as Deep SORT, are available. We are currently modifying a version of Deep SORT to handle not only multiple participants but to avoid confusion when someone turns their head or looks away, thinking they are now someone else. We have plans to collect multiple participant data once it is working.
- Our real-time component requires per-person calibration at the beginning of a session. This is time-consuming and provides a poor user-experience but necessary given different ambient conditions. A more user-friendly approach is needed.
- The current training data was collected in a laboratory and not in a classroom where different room temperature characteristics and ventilation conditions could be expected (e.g., fans running in the classroom blowing air towards the peer group). A broader and much larger dataset, with more participants and varied collection environments, is necessary for full validation of our approach. We have plans to collect such data.
- The data collected for training was only from adults and not the targeted middle school students due to privacy concerns yet

school-age children’s thermal properties are likely to differ from those of adults. The face/mouth/nostril region detection was computed on individually-stored thermal video frames in a separate detection phase after the data collection phase to preserve the 25 fps thermal camera capture rate. The addition of a Raspberry Pi AI module could increase the frame rate substantially for real-time detection so that no potentially personally-identifiable information (namely the thermal video frames of a student) would need to be retained. Instead, only the thermal temperature data and the coordinates of the face/mouth/nostril regions would need to be stored for use in calculating the features used for training the classifier. Similarly, the real-time thermal VAD/engagement system could perform all calculations without retaining the thermal video frames. We are testing the Raspberry Pi AI module and have had great results for face/mouth/nostril detection achieving 25 fps. However, the AI module does not speed up the trained classifier limiting the system to 5 fps.

- Currently, we have only employed our Visual-Only_RF model as a baseline but recognize another baseline could provide valuable insight. We tested TalkNet as a baseline and found it does not run in real-time and cannot detect facial regions in thermal false-color imagery, as the audio-visual synchrony model relies on visible-spectrum lip texture absent in thermal imaging. Future work will add a parallel RGB camera to enable a proper audio-visual comparison on aligned video footage to provide another baseline.
- We currently only explore a laboratory setup as proof-of-concept. We will perform additional validation in more realistic settings once the full iSAT pipeline is tested.

Acknowledgments

I would like to acknowledge the Brandeis iSAT team for helpful comments on my draft. This research was conducted as a volunteer on the NSF AI Institute for Student AI Teaming (iSAT) grant, NSF DRL Award 2454151. *AI assistance disclosure: Claude AI, ChatGPT, and Google Gemini assisted in software development and editing this document to fit page limitations.*

References

- M. Abdrakhmanova, A. Kuzdeuov, S. Jarju, Y. Khasanov, M. Lewis, and H. A. Varol. 2021. *Speaking-Faces: A Large-Scale Multimodal Dataset of Voice Commands with Visual and Thermal Video Streams*. *Sensors*, 21(10).
- R. S. Baker and J. Ocumpaugh. 2014. *Cost-effective, actionable engagement detection at scale*. In *Educational Data Mining*.
- D. Bennett, S. Cahlan, and D. Taylor. 2020. *Military-grade camera shows risks of airborne coronavirus spread*. *Washington Post* (December 11, 2020).
- N. Bosch, S. D’Mello, J. Ocumpaugh, R. Baker, and V. Shute. 2016. *Using video to automatically detect learner affect in computer-enabled classrooms*. *ACM Transactions on Interactive Intelligent Systems*, 6.
- H. Cao, Š. Beňuš, R. C. Gur, R. Verma, and A. Nenkova. 2014. *Prosodic cues for emotion: Analysis with discrete characterization of intonation*. In *Proceedings of Speech Prosody 2014*, pages 130–134.
- J. Cao, A. Ganesh, J. Cai, R. Southwell, E. M. Perkoff, M. Regan, K. Kann, J. H. Martin, M. Palmer, and S. D’Mello. 2023. *A comparative analysis of automatic speech recognition errors in small group classroom discourse*. In *Proceedings of UMAP 2023*, pages 1–13.
- K. Y. Chan, B. Abu-Salih, R. Qaddoura, A. M. Al-Zoubi, V. Palade, D.-S. Pham, J. Del Ser, and K. Muhammad. 2023. *Deep neural networks in the cloud: Review, applications, challenges and research directions*. *Neurocomputing*, 545:126327.
- S. D’Mello and A. Graesser. 2012. *Dynamics of affective states during complex learning*. *Learning and Instruction*, 22(2):145–157.
- B. A. Goodman, J. Drury, R. D. Gaimari, L. Kurland, and J. Zarrella. 2006. *Applying user models to improve team decision making*. Technical Report MTR 060150, MITRE Corporation, Bedford, MA.
- Y. Guan, V. A. Trinh, V. Voleti, and J. Whitehill. 2024. *Multi-modal speech transformer decoders: When do multiple modalities improve accuracy?* *arXiv:2409.09221*.
- Y. Guan, V. A. Trinh, V. Voleti, and J. Whitehill. 2025. *MLLM-based speech recognition: When and how is multimodality beneficial?* *arXiv:2507.19037*.
- A. Guryanov. 2019. *Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees*. In *Analysis of Images, Social Networks and Texts: AIST 2019*, pages 39–50.
- S. Ioannou, V. Gallese, and A. Merla. 2014. *Thermal infrared imaging in psychophysiology: Potentialities and limits*. *Psychophysiology*, 51(10):951–963.
- J. Kwon, O. Kwon, K. Taek Oh, J. Kim, and S. K. Yoo. 2023. *Breathing-Associated Facial Region Segmentation for Thermal Camera-Based Indirect Breathing Monitoring*. *IEEE Journal of Translational Engineering in Health and Medicine*, 11:505–514.
- A. Liaw and M. Wiener. 2002. *Classification and regression by randomForest*. *R Journal*, 2(3).
- M. C. T. Manullang, Y.-H. Lin, S.-J. Lai, and N.-K. Chou. 2021. *Implementation of thermal camera for non-contact physiological measurement: A systematic review*. *Sensors*, 21(23):7777.
- A. Meister, S. Büchner, and A. Amthor. 2015. *State machine based nonlinear hysteresis model*. *Mechanics*, 31:215–221.
- S. M. Meyers, K. Kisling, T. F. Atwood, and X. Ray. 2022. *A standardized workflow for respiratory-gated motion management decision-making*. *Journal of Applied Clinical Medical Physics*, 23(8):e13705.
- W. S. Noble. 2006. *What is a support vector machine?* *Nature Biotechnology*, 24(12):1565–1567.
- M. Oztoprak. 2022. *Engagement detection using prosodic cues*. Master’s thesis, Northern Illinois University.
- D. Palmer, Y. Zhu, K. Lai, H. VanderHoeven, M. Bradford, I. Khebour, C. Mabrey, J. Fitzgerald, N. Krishnaswamy, M. Palmer, and J. Pustejovsky. 2025. *Speech is not enough: Interpreting nonverbal indicators of common knowledge and engagement*. In *Proceedings of AAAI 2025*, 39(28):29676–29678.
- P. Sharma, J. Shubham, S. Gautam, S. Maharjan, S. R. Khanal, M. C. Reis, J. Barroso, and V. M. J. Filipe. 2023. *Student engagement detection using emotion analysis, eye tracking and head movement with machine learning*. *arXiv:1909.12913*.
- Silero Team. 2024. *Silero VAD: Pre-trained enterprise-grade voice activity detector*. GitHub repository.
- R. Tao, Z. Pan, R. K. Das, X. Qian, M. Z. Shou, and H. Li. 2021. *Is someone speaking? Exploring long-term temporal features for audiovisual active speaker detection*. In *Proceedings of ACM Multimedia 2021*, pages 1–10.
- N. Tran, D. Litman, B. Pierce, R. Correnti, and L. C. Matsumura. 2025. *Improving in-context learning example retrieval for classroom discussion assessment*. In *Proceedings of BEA 2025*.
- H. VanderHoeven, N. Blanchard, and N. Krishnaswamy. 2023. *Robust motion recognition using gesture phase annotation*. In *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*, pages 592–608. Springer.
- N. S. Vemulapalli, P. Paladugula, G. S. Prabhat, S. Abhishek, and T. Anjali. 2023. *Face detection with landmark using YOLOv8*. In *Proceedings of ICEFEET 2023*, pages 1–5. IEEE.

- J. Wang, S. Dudy, X. He, Z. Wang, R. Southwell, and J. Whitehill. 2024. *Speaker diarization in the classroom: How much does each student speak in group discussions?* In *Proceedings of EDM 2024*, pages 360–367.
- J. Wang, S. Dudy, X. He, Z. Wang, R. Southwell, and J. Whitehill. 2025. *Optimizing speaker diarization for the classroom*. *Journal of Educational Data Mining*, 17(1):98–125.
- A. N. Wilson, K. A. Gupta, B. H. Koduru, A. Kumar, A. Jha, and L. R. Cenkeramaddi. 2023. *Recent advances in thermal imaging and its applications using machine learning: A review*. *IEEE Sensors Journal*, 23(4):3395–3407.
- N. Wojke, A. Bewley, and D. Paulus. 2017. *Simple online and realtime tracking with a deep association metric*. *arXiv:1703.07402*.
- X. Xu, D. M. Dugdale, X. Wei, and W. Mi. 2023. *Leveraging artificial intelligence to predict young learner online learning engagement*. *American Journal of Distance Education*, 37(3):185–198.
- T. Yoshida, E. Yamazaki, and S. Hangai. 2008. *Spoken Word Recognition from Side of Face Using Infrared Lip Movement Sensor*. In E. André et al. (eds.), *Perception in Multimodal Dialogue Systems*, Lecture Notes in Computer Science, vol. 5078, pages 1–6. Springer.
- Y. Zhu, I. Dey, K. Lai, and J. Pustejovsky. 2026. *Nonverbal Behavior Recognition in Multimodal Interactions from RGB Video*. In A. Lücking and A. Mehler (eds.), *Behavioromics*. Springer.

A Appendix

A.1 Updated Tables and Figure for Revised Feature Set

Table 5 shows permutation feature importance for the revised 12-feature model after 12-frame VAD label correction; *NM_Xcorr* (+0.030) and *Mouth_Breath_BRSI* (+0.028) rise to rank 1–2, with 11 of 12 features positive, validating thermal channel decoupling as the dominant physiological marker of speech. Table 6 defines this revised 12-feature thermal-only set, replacing the original 125-frame rolling averages and redundant instantaneous temperature columns with slope-based dynamics and spectral coupling features. Figure 4 shows one sample result applying the revised 12-feature model to detect speech and quiet segments in real time. A per-session logistic recalibration layer, fitted on a short calibration sequence and applied with baseline correction for slow-responding breathing features, compensates for inter-subject variability in the RF classifier and achieves an ECI of 0.724 with mean speech-offset detection latency of 3.4 seconds on this example session. Much more testing will be required to determine if this holds up in practice with different subjects and ambient conditions.

Feature	Importance	Rank
<i>NM_Xcorr</i>	+0.0304	1
<i>Mouth_Breath_BRSI</i>	+0.0281	2
<i>Breath_BRSI</i>	+0.0096	3
<i>Center_Mouth_Contrast</i>	+0.0058	4
<i>Mouth_to_Face_Delta</i>	+0.0044	5
<i>Nostril_to_Face_Delta</i>	+0.0031	6
<i>Mouth_Slope_Accel</i>	+0.0010	7
<i>Slope_Coupling</i>	+0.0010	8
<i>Mouth_Temp_Max_Slope</i>	+0.0006	9
<i>Airflow_Dominance</i>	+0.0004	10
<i>Face_Temp_Spread</i>	+0.0002	11
<i>Nostril_Temp_Max_Slope</i>	−0.0048	12

Table 5: Permutation feature importance for the revised 12-feature Thermal-Only RF model (18 LOEO folds, 12-frame VAD label correction applied). Importance is mean F1 decrease when feature is permuted. Compare with Table 3 (original 19-feature model).

Feature	Type	Signal captured	Speech vs. quiet
<i>Thermal airflow gradients</i>			
<i>Mouth_to_Face_Delta</i>	Thermal	Z-scored mouth temp minus mean face temp	Higher during speech
<i>Nostril_to_Face_Delta</i>	Thermal	Z-scored nostril temp minus mean face temp	Lower during speech
<i>Face_Temp_Spread</i>	Thermal	Max–min temperature across face ROI (°C)	Higher during speech
<i>Center_Mouth_Contrast</i>	Thermal	Center-mouth / mean-mouth temperature ratio	Higher during speech
<i>Airflow velocity dynamics</i>			
<i>Mouth_Temp_Max_Slope</i>	Thermal	Max $ dT/dt $ mouth ROI, 100-frame window	Higher at speech onset
<i>Nostril_Temp_Max_Slope</i>	Thermal	Max $ dT/dt $ nostril ROI, 100-frame window	Lower during speech
<i>Airflow_Dominance</i>	Thermal	$ S_m / (S_m + S_n + \varepsilon)$	Approaches 1 in speech
<i>Slope_Coupling</i>	Thermal	$\min(S_m , S_n) / (\max(S_m , S_n) + \varepsilon)$	Low during speech
<i>Mouth_Slope_Accel</i>	Thermal	Second derivative of mouth temperature	Peaks at onset/offset
<i>Breathing rhythm and spectral coupling</i>			
<i>Breath_BRSI</i>	Spectral	Nostril rhythm stability index (100-frame FFT)	Lower during speech
<i>Mouth_Breath_BRSI</i>	Spectral	Mouth rhythm stability index (100-frame FFT)	Lower during speech
<i>NM_Xcorr</i>	Spectral	Nostril–mouth temperature Pearson correlation	Decouples during speech

Table 6: Revised 12-feature thermal-only set used in the updated classifier. S_m, S_n : maximum temperature slopes in mouth and nostril ROIs; ε : stability constant. Z-scoring of the 2 delta features (normalized by a per-session mean and standard deviation, then clipped to ± 3) helps handle session-to-session room temperature variation. Other features like *Face_Temp_Spread* and the slope features might also benefit from z-scoring. *MAR_Variance* (visual) is excluded from the Thermal-Only RF but included in the Fusion RF variant (13 features, F1 = 0.8267).

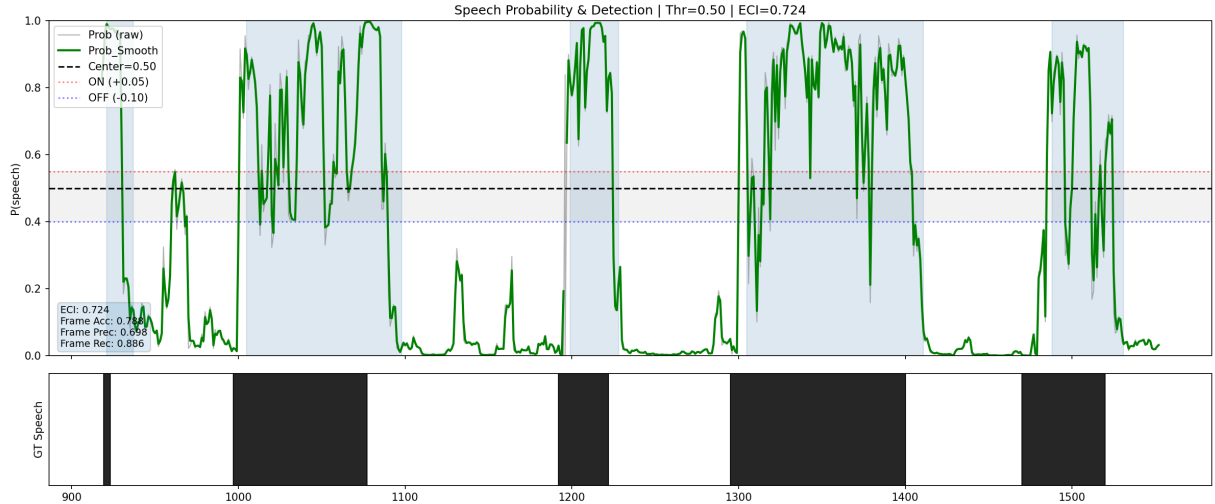


Figure 4: Real-time speech probability trajectories for a single subject using 12-feature thermal-only model. ECI=0.724; Episode Recall=1.0; Frame Precision=0.698; Frame Recall=0.886. Gray: raw RF probability output; green: 5-frame smoothed probability; black dashed: calibrated threshold T_{base} ; red/blue dotted: hysteresis ON/OFF boundaries; light blue shading: predicted speaking intervals. Ground truth (GT) shown below (black = speaking, white = quiet).

Investigating Context-aware CTC for Pronunciation Assessment: Mitigating Peak Behavior and Context Independency Assumption

Jiun-Ting Li¹ Bi-Cheng Yan² Tien-Hong Lo²
Shih-Hsuan Chiu³ Fu-An Chao² Berlin Chen²

¹Advanced Technology Laboratory, Chunghwa Telecom Co., Ltd., Taiwan

²National Taiwan Normal University, Taipei, Taiwan

³National Taiwan University, Taipei, Taiwan

jtli@cht.com.tw

Abstract

Automatic pronunciation assessment (APA) provides L2 learners with scalable and timely feedback on pronunciation proficiency in a target language, typically through goodness of pronunciation (GOP) features. GOP quantifies how well a pronounced phoneme matches the expected target sound by comparing acoustic features against the model's posterior probabilities. Traditional GOP relies on forced alignment to obtain these posteriors, but it suffers from acoustic-induced misalignments that degrade assessment reliability. Although the standard CTC-GOP approach bypasses forced alignment, it is limited by the inherent peaky behavior of CTC-based ASR models, which produces sparse posteriors and lacks stable temporal information. To address these issues in standard CTC, we propose a context-aware CTC framework incorporating output context dependency (OCD) in the CTC topology, along with label prior (LP) and maximum conditional entropy (EnCTC) regularization, to mitigate peakiness and produce more stable ASR logits suitable for GOP computation. Experiments on the speechocean762 corpus demonstrate that our best context-aware configurations achieve superior phoneme-level performance, outperforming the TDNN-F baseline and standard CTC in unified GOPT (phoneme PCC 0.641 vs. 0.612; word total PCC 0.582 vs. 0.549) while narrowing the gap in hierarchical HierCB scoring. These improvements widen the scoring margin between correct and mispronounced phonemes from 0.708 to 0.816 in GOPT. They also reveal that mitigating CTC peakiness and incorporating context dependency significantly enhance CTC-GOP stability and robustness, especially for alignment-free APA models.

1 Introduction

Goodness of pronunciation (GOP) (Witt and Young, 2000; Hu et al., 2015) aims to evaluate whether pronunciation is in the correct way (mispronunciation detection and diagnosis, MDD) (Parikh et al.,

2025) and assess their pronunciation proficiency (automatic pronunciation assessment, APA) (Zhang et al., 2021; Chao et al., 2022; Gong et al., 2022; Do et al., 2023; Pei et al., 2024; Yan et al., 2024, 2025). The conventional GOP uses the hidden Markov model, operating force alignment to retrieve the time mapping with logits and the given phoneme sequence, then computing the similarity with aligned phonemes and logits (Zhang et al., 2021). The GOP score and feature have been widely adopted in MDD and APA tasks.

Despite the success, the conventional GOP has a major drawback. It depends on accurate audio-text alignments. Any misalignment degrades assessment quality. To solve this issue, an alignment-free method, named CTC-GOP (Cao et al., 2024), was proposed. CTC-GOP is based on connectionist temporal classification (CTC) (Graves et al., 2006)-based automatic speech recognition (ASR) models, utilizing path-wise decoding results to compute GOP scores and features. It does not need the force alignment procedure, avoiding the misalignment issues.

However, despite its significant contributions, CTC-GOP is not without its limitations. While traditional DNN-HMM systems are precise in timing but often too rigid to accommodate the disfluencies of L2 speech. In contrast, standard CTC is highly flexible but suffers from inherent peaky behavior, where the model strongly prioritizes what was said over when it occurred. Since accurate temporal information is essential for APA tasks, this remains a critical drawback. In detail, the CTC topology introduces a special blank token to act as a symbol boundary. During training, the ASR model over-identifies blank tokens, reinforcing this tendency until most frames are classified as "blank" token (Zeyer et al., 2021). Consequently, the probability distributions of non-peaky frames are underestimated, with optimal phoneme distributions restricted to only a few sharp spikes, making it

difficult to handle pronunciation uncertainty in ambiguous pronunciation. While this peaky behavior and the resulting ambiguity in word boundaries do not hinder general ASR tasks, this lack of temporal density degrades the efficacy of CTC-GOP, which relies on stable frame-level posteriors.

Inspired by content-aware GOP (Shi et al., 2020), we explore two methods to alleviate peakiness and improve the accuracy of token onset and offset predictions: (1) a label prior (LP) to the CTC loss to penalize blank-heavy paths (Huang et al., 2024), and (2) maximum conditional entropy-based regularization for CTC (EnCTC), which can be conducted without frame-level supervision due to the lack of labels for phoneme boundaries (Chen et al., 2023). These adjustments aim to produce more informative, less peaky posteriors suitable for phonetic modeling. Furthermore, CTC primarily stems from the core assumptions of context independence. Although modern end-to-end (E2E) architectures, such as TDNN-F (Peddinti et al., 2015; Povey et al., 2018) and attention-based models (Baevski et al., 2020; Hsu et al., 2021), incorporate input context dependency (ICD) through contextualized representations, the standard CTC topology remains constrained by a strong conditional independence assumption. In natural speech, continuous phonemes exhibit significant coarticulation, suggesting that modeling dependencies between preceding and succeeding units may be superior to the standard independence model. To address this, we use context dependency (CD) symbols in CTC topology (Chorowski et al., 2019), referred to as output context dependency (OCD), to investigate the implications of the OCD impact on APA.

After these attempts to fix CTC, we utilize the resulting ASR models to compute CTC-GOP features for two distinct scoring architectures: the unified GOPT (Gong et al., 2022) and the hierarchical HierCB (Yan et al., 2025). Because GOPT maps phoneme-level CTC-GOP features directly to scores, serving as an alignment-free APA model. In contrast, HierCB incorporates additional inputs that require explicit time alignment (TA) information, such as phoneme-level acoustic embeddings and duration features, serving as an alignment-dependent APA model.

To sum up our contributions: (1) We develop a context-dependent CTC framework that incorporates OCD and regularization techniques to mitigate the inherent peaky behavior of standard CTC. (2) We systematically analyze how architectural

choices, such as OCD, label priors, and stress markers, impact time alignment (TA) accuracy across multiple speech corpora. (3) We evaluate these improvements across both unigram and hierarchical APA architectures, demonstrating that mitigating CD is a factor for achieving trustworthy scoring results in alignment-dependent assessment models.

2 Methodology

2.1 CTC training with LP

ASR task aims to predict the most probable token sequence $\hat{Y}$. Given a speech feature matrix X , whether it can be raw speech or a sequence of feature vectors, is conveyed to an audio encoder. It processes X into a sequence of contextualized hidden features. Subsequently, it can be applied to MLP (Popescu et al., 2009), or CD embedding layers (Chorowski et al., 2019), then culminating in a softmax activation function. This process yields the probability distribution over each token in $\hat{Y}$. CTC models the posterior probability $P(Y|X)$ by marginalizing all possible alignments $\mathcal{B}^{-1}(Y)$ between X and Y with the conditional independence assumption.

$$P(Y|X) = \sum_{\pi \in \mathcal{B}^{-1}(Y)} \prod_{t=1}^T p_t(\pi_t|X), \quad (1)$$

where $\mathcal{B}(\pi)$ collapses an alignment sequence π , merging repeated neighbouring tokens and then removing any blank labels. $p_t(\pi_t|X)$ is the posterior probability of token at time t .

To reduce the number of blanks in the optimal alignment paths, we can apply unigram LPs on each token in π to penalise paths containing too many blank symbols.

$$P^{LP}(Y|X) = \sum_{\pi \in \mathcal{B}^{-1}(Y)} \prod_{t=1}^T \frac{p_t(\pi_t|X)}{P(\pi_t)^a}, \quad (2)$$

where the hyper-parameter $a \in \mathbb{R}$ is a scalar. When $a = 0$, this is exactly the standard CTC loss. If π has more blank tokens, it will have a larger prior $P(\pi_t)$, in other words, get a larger penalty in its posterior probability, so $\mathcal{B}^{-1}(Y)$, including the optimal one, will avoid blank tokens. Notably, the LPs are applied during training. And if using LP while training, there is no further need to apply LPs while decoding.

2.2 EnCTC

While LP ensures the ASR model is fair to rare symbols, EnCTC ensures the ASR model is cautious and doesn't overfit to its own predictions. We utilize EnCTC to prevent the ASR model from converging prematurely to a single, overconfident alignment path. This is achieved by adding a maximum conditional entropy term to the standard CTC objective:

$$\mathcal{L}_{\text{EnCTC}} = \mathcal{L}_{\text{CTC}} - bH(\pi|Y, X), \quad (3)$$

where b regulates the intensity of the regularization. The entropy $H(\cdot)$ measures the uncertainty across all feasible alignment paths. By maximizing this entropy, the ASR model is encouraged to distribute probability mass across multiple nearby paths, leading to less peaky posteriors and more informative frame-level distributions for GOP computation.

2.3 DWFST implementation for standardized CTC

To benefit from HMM's highly accurate time alignment (TA), and simultaneously imitate the CTC, which can be considered to have a one-state-HMM topology with a single self-loop. In literature, it can easily be done by defining a topology (Zhao and Bell, 2024), integrated into an E2E ASR framework with a differentiable weighted finite state transducer (DWFST) (Hannun et al., 2020). The posterior $P(Y|X)$ can be rewritten as

$$P(Y|X) = \sum_{\pi \in \Pi(Y; \mathbf{T} \circ \mathbf{L})} p(\pi|X), \quad (4)$$

where $\mathbf{T}$ and $\mathbf{L}$ denote the topology and the lexicon finite-state transducer (FST), and $\Pi(Y; \mathbf{T} \circ \mathbf{L})$ represents the set of all the token sequences, whose corresponding output sequence is Y , in the composed FST, $\mathbf{T} \circ \mathbf{L}$.

To compute the score of each probable π , we first need to intersect the decoding graph, which is $\mathbf{T} \circ (\mathbf{L} \circ \mathbf{G})$, where $\mathbf{G}$ is the target word sequence, with the emission FST constructed $\mathbf{E}$ from the ASR model's logit outputs, which is like

$$\sum_{\pi \in \Pi(Y; \mathbf{T} \circ \mathbf{L})} p(\pi|X) = \text{TotalScore}(\mathbf{E} \circ (\mathbf{T} \circ (\mathbf{L} \circ \mathbf{G}))). \quad (5)$$

2.4 Decoding with alignment graph built without lexicon

The Viterbi decoding does not guarantee the alignment output to be identical to the canonical phone sequence, even if the same word sequence is output. To solve this problem, we follow the idea in Zhang et al. (2021) to build the lexicon-to-grammar FST directly using the provided canonical phone sequence. We first create a linear FST in which input labels are the canonical phone sequences, and the output labels are the corresponding words and epsilons. The disambiguation symbol is added at the tail of LG to ensure the determinization of FST.

2.5 CTC training with context-dependent symbols

To address the limitations of context-independent targets, we transition from monophone to diphone units by following the idea in Chorowski et al. (2019). For a target sequence such as $[q, v, v]$, the corresponding extended transcript is mapped to $[\emptyset q, qv, vv]$, where $\emptyset q$ and qv are variant units for q unit, $\emptyset$ is the blank token, following the alignment constraints:

$$\emptyset^* \emptyset q \emptyset q^* \emptyset^* qv qv^* \emptyset^* vv vv^* \emptyset^*,$$

While this expansion captures local phonetic context, it significantly increases the output dimensionality and computational complexity. To mitigate this, we adopt a context-dependent embedding network to replace the standard final linear layer. This approach generates co-dependent prototype vectors for the diphone units, effectively reducing parameter overhead while enabling the ASR model to generalize across unseen phonetic contexts.

2.6 CTC-GOP computation

The definition of CTC-GOP adopts the approach of enumerating all possible transcription paths involving deletions and substitutions of specific tokens. We denote the canonical phoneme sequence as Y_c . In place of the original GOP-NN (Hu et al., 2015), this method first defines the log posterior probability (lpp) scalar as

$$lpp = \log p(Y_c|X). \quad (6)$$

The vector of log posterior ratios (**LPR**) for the i th phoneme in Y_c is then defined as

$$\mathbf{LPR}(i) = \log \left(\frac{p(Y_c|X)}{p(Y_{sdi}|X)} \right), \quad (7)$$

where the Y_{sdi} is the deletion and substitutions for the i th phoneme in Y_c . The features of the i th phoneme in Y_c is then defined to be:

$$\text{CTC-GOP}(i) = \{lpp, \mathbf{LPR}(i)\}. \quad (8)$$

Regarding the dimensionality of $\text{CTC-GOP}(i)$, for a 39-phoneme set, the $\mathbf{LPR}$ vector consists of 40 dimensions (accounting for 38 possible substitutions, one deletion, and the canonical phoneme itself). Including the lpp scalar, the total feature vector length is 41.

We also describe the details for how to extract CTC-GOP features from a CTC-based ASR model in Appendix A. The pipeline consists of three stages: (1) reduction of the phone inventory from full phone sets to pure phones, (2) batched CTC loss computation over substitution and deletion hypotheses, and (3) construction of $\mathbf{LPR}$.

3 Experimental Results

3.1 Corpus

We fine-tune the ASR models on the 960-hour LibriSpeech (Panayotov et al., 2015) corpus using a 72-phoneme inventory (including stress markers). To evaluate TA accuracy, we utilize two standard American English corpora with gold-standard phonetic alignments: TIMIT (Garofolo et al., 1993) (read speech) and Buckeye (Pitt et al., 2005) (conversational speech).

Our evaluation set includes 400 dev and 192 test samples from TIMIT, totaling 177,080 phonemes. For Buckeye, we evaluate 7,492 samples containing 858,386 phonemes. These datasets provide the verbatim timed transcriptions necessary to measure the impact of mitigated peaky behavior on boundary precision. Detailed hyperparameters and implementation are provided in Appendix B.

Both TIMIT (61 pure phones) and Buckeye (76 pure phones) were standardized to the CMU 39-pure phone units. This ensures a uniform phonetic label space across the three corpora, regardless of their original inventory sizes. Then, we add the stress markers back to their phoneme sequences L_{ref} . We first use the TDNN-F model pretrained on LibriSpeech¹ to transcribe the phoneme sequence L_{hyp} . Then, we apply a positional transfer function to map the possible stress markers from transcription to the converted non-stress labels, by

identifying vowel indices V_{ref} and V_{hyp} and applying a conditional mapping strategy:

- Linear mapping: If $|V_{ref}| = |V_{hyp}|$, stress markers are transferred via a direct 1-to-1 ordinal correspondence in L_{ref} and L_{hyp} .
- Relative alignment: If vowel counts diverge, we apply a nearest-neighbor approach based on relative sequence position. For each V_j^{ref} in L_{ref} , the stress digit is borrowed from the V_j^{hyp} in L_{hyp} that minimizes the normalized distance: $\hat{i} = \arg \min_i \left| \frac{i}{|L_{ref}|} - \frac{j}{|L_{hyp}|} \right|$

This mechanism ensures that the hypothesis sequence maintains a rhythmic structure consistent with the reference, even in the presence of phoneme insertions or deletions.

In addition to investigating the effect on acoustic variants, we evaluate ASR models trained on two distinct phoneme inventories: a comprehensive 72-unit set and a reduced 39-unit set, to provide a controlled benchmark against the original set.

For APA evaluation, we conducted experiments on the speechocean762 corpus (Zhang et al., 2021), a publicly available dataset specifically designed for APA research. This dataset contains 5,000 English-speaking recordings spoken by 250 Mandarin L2 learners. Each of the training and test sets has 2,500 utterances, where annotates utterance-level scores (accuracy, stress, completeness, fluency, prosodic, total), word-level scores (accuracy, stress, total) and phoneme accuracy score.

3.2 Backbone ASR models

WavLM (Chen et al., 2022) improves upon wav2vec 2.0 (Baeovski et al., 2020) and HuBERT (Hsu et al., 2021), enhancing the model’s robustness for APA by incorporating gated relative position bias and speech denoising capabilities during pre-training. While it is similar to HuBERT, improving upon wav2vec 2.0 by predicting hidden unit clusters, WavLM’s unique focus on simulating noisy and overlapped speech during training allows for more discriminative acoustic representations. These features help capture the small phonetic changes needed for accurate, high-resolution pronunciation scores.

3.3 Metrics

To access TA quality at the word- and phoneme-level, we adopt both absolute and marginal metrics

¹<https://kaldi-asr.org/models/m13>

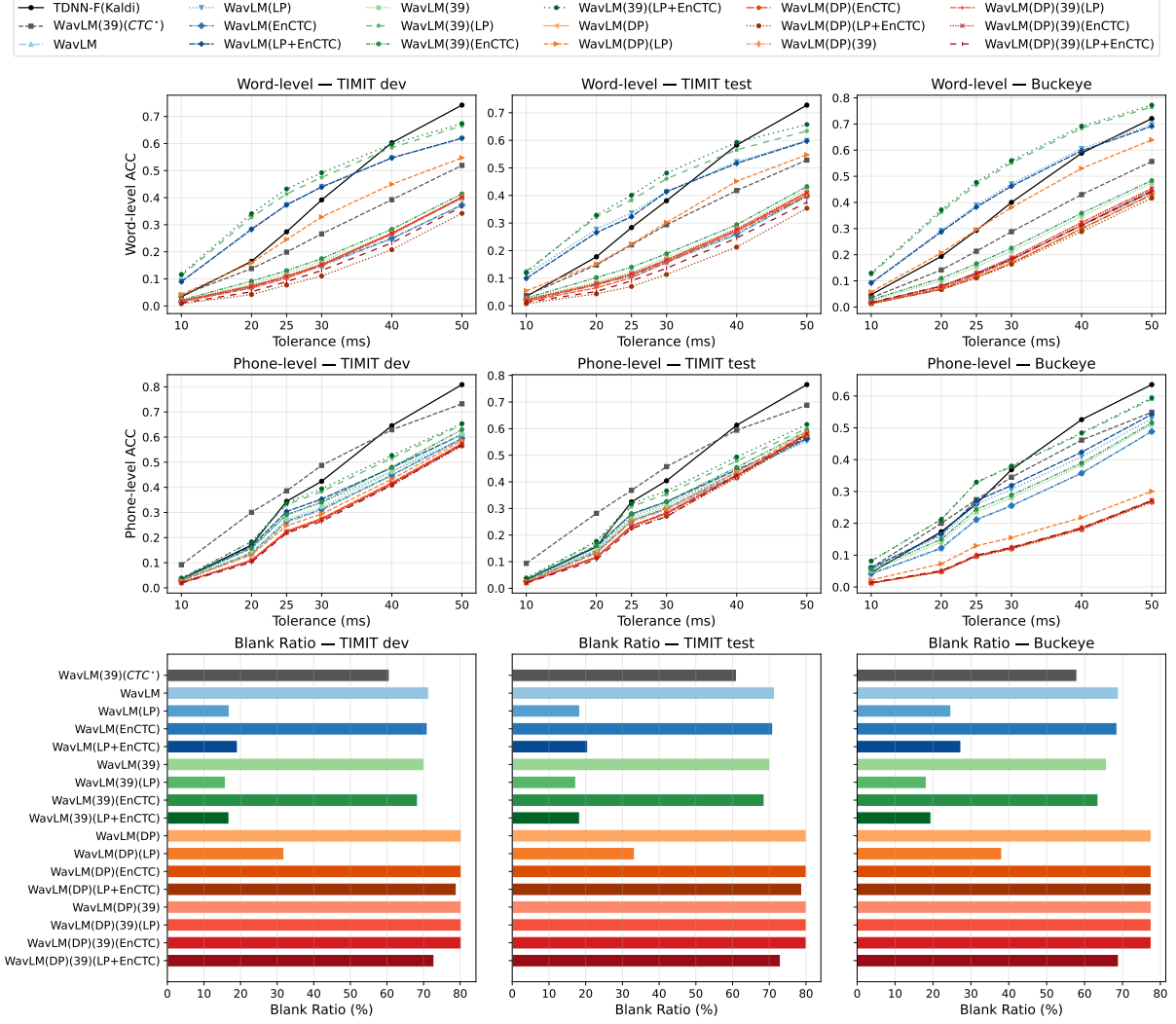


Figure 1: The time stamp error (TSE), alignment accuracy (ACC) and blank ratio (BR) figures. These ASR models test on TIMIT’s dev and eval dataset and the whole Buckeye corpus.

(Zhao and Bell, 2024). For the absolute one, we use the time stamp error (TSE), which is defined as:

$$\text{TSE} = \frac{\sum_w |r_s - h_s| + |r_e - h_e|}{N_w}, \quad (9)$$

where the N_w represents the phoneme or word count in each utterance, and r_s , h_s , r_e , h_e indicate the reference, hypothesis’ start time and end time, respectively. Another one is alignment accuracy (ACC) for a threshold τ milliseconds, $\tau = \{10, 20, 25, 30, 40, 50\}$, which is defined:

$$\text{ACC}(\tau) = \frac{\sum_w \mathbf{1}(r_s - \tau \leq h_s \cap h_e \leq r_e + \tau)}{N_w}, \quad (10)$$

for each phoneme or word that satisfies the condition, $\mathbf{1}(\star)$ outputs 1, otherwise 0.

To estimate the efficacy of those methods in mitigating CTC’s peaky behavior, we also count the blank ratio (BR) to understand whether it decreases with those methods. BR is defined as

$$\text{BR} = \frac{\sum_u N_b}{\sum_u N}, \quad (11)$$

where N_b denotes the number of occurrences of the blank state within an evaluation utterance, and N represents the total number of frames in that utterance. The summation $\sum_u$ is executed across all evaluation utterances. This evaluation metric allows us to quantify the proportion of frames that are classified as blank states by our models.

To evaluate the APA model’s performance, the Pearson correlation coefficient (PCC) is adopted, while the mean squared error (MSE) is only used for the phoneme accuracy.

ASR Models	Phoneme Score		Word Score (PCC)			Utterance Score (PCC)				
	MSE ↓	PCC ↑	Acc. ↑	Stress ↑	Total ↑	Acc. ↑	Comp. ↑	Fluency ↑	Prosodic ↑	Total ↑
TDNN-F(Kaldi) (Gong et al., 2022)	0.085 ±0.001	0.612 ±0.003	0.533 ±0.004	0.291 ±0.030	0.549 ±0.002	0.714 ±0.004	0.155 ±0.039	0.753 ±0.008	0.760 ±0.006	0.742 ±0.005
WavLM(39)(CTC*) (Cao et al., 2024, 2026)	0.102 ±0.003	0.484 ±0.019	0.443 ±0.014	0.112 ±0.021	0.449 ±0.019	0.710 ±0.002	0.142 ±0.090	0.661 ±0.004	0.699 ±0.002	0.719 ±0.002
WavLM (DWFST-CTC)										
-	0.086 ±0.002	0.612 ±0.014	<u>0.554</u> ±0.014	0.306 ±0.027	<u>0.573</u> ±0.013	0.742 ±0.006	0.155 ±0.048	0.745 ±0.005	0.749 ±0.004	0.762 ±0.006
(LP)	0.123 ±0.001	0.335 ±0.005	0.347 ±0.002	0.032 ±0.005	0.356 ±0.000	0.621 ±0.008	-0.013 ±0.036	0.655 ±0.015	0.643 ±0.009	0.622 ±0.011
(EnCTC)	0.089 ±0.002	0.590 ±0.009	<u>0.541</u> ±0.006	0.254 ±0.010	<u>0.559</u> ±0.007	<u>0.729</u> ±0.007	<u>0.179</u> ±0.033	0.739 ±0.003	0.743 ±0.001	<u>0.745</u> ±0.003
(LP+EnCTC)	0.120 ±0.001	0.352 ±0.012	0.365 ±0.006	0.043 ±0.025	0.379 ±0.006	0.641 ±0.008	-0.007 ±0.012	0.670 ±0.014	0.659 ±0.013	0.654 ±0.011
(39)	0.088 ±0.002	0.600 ±0.010	<u>0.547</u> ±0.008	<u>0.303</u> ±0.005	<u>0.565</u> ±0.009	<u>0.725</u> ±0.006	<u>0.174</u> ±0.011	0.740 ±0.006	0.745 ±0.003	<u>0.749</u> ±0.006
(39)(LP)	0.125 ±0.000	0.319 ±0.000	0.336 ±0.004	0.022 ±0.002	0.347 ±0.001	0.594 ±0.006	0.033 ±0.033	0.621 ±0.013	0.615 ±0.010	0.601 ±0.008
(39)(EnCTC)	0.122 ±0.000	0.325 ±0.001	0.336 ±0.002	0.016 ±0.013	0.347 ±0.003	0.607 ±0.003	0.022 ±0.014	0.641 ±0.006	0.635 ±0.002	0.619 ±0.006
(39)(LP+EnCTC)	0.089 ±0.002	0.595 ±0.012	<u>0.542</u> ±0.011	<u>0.299</u> ±0.021	<u>0.560</u> ±0.011	<u>0.728</u> ±0.008	0.230 ±0.046	0.736 ±0.005	0.744 ±0.004	<u>0.747</u> ±0.006
(DP)	<u>0.084</u> ±0.001	<u>0.621</u> ±0.004	<u>0.549</u> ±0.004	0.210 ±0.012	<u>0.567</u> ±0.003	<u>0.724</u> ±0.002	<u>0.219</u> ±0.050	0.720 ±0.005	0.732 ±0.003	0.739 ±0.004
(DP)(LP)	0.106 ±0.001	0.486 ±0.012	0.444 ±0.012	0.134 ±0.026	0.463 ±0.010	0.660 ±0.005	0.094 ±0.019	0.659 ±0.004	0.664 ±0.001	0.675 ±0.004
(DP)(EnCTC)	<u>0.083</u> ±0.001	<u>0.626</u> ±0.003	<u>0.553</u> ±0.004	0.227 ±0.010	<u>0.571</u> ±0.003	<u>0.727</u> ±0.004	<u>0.228</u> ±0.077	0.726 ±0.009	0.739 ±0.006	<u>0.745</u> ±0.002
(DP)(LP+EnCTC)	0.096 ±0.002	0.558 ±0.006	0.511 ±0.005	0.159 ±0.022	0.527 ±0.004	0.704 ±0.005	0.061 ±0.046	0.702 ±0.008	0.711 ±0.014	0.718 ±0.007
(DP)(39)	0.081 ±0.001	0.641 ±0.005	0.565 ±0.005	0.261 ±0.018	0.582 ±0.004	<u>0.728</u> ±0.002	<u>0.220</u> ±0.088	0.726 ±0.003	0.736 ±0.002	<u>0.747</u> ±0.003
(DP)(39)(LP)	<u>0.081</u> ±0.001	<u>0.640</u> ±0.006	<u>0.564</u> ±0.006	0.262 ±0.016	<u>0.582</u> ±0.006	<u>0.729</u> ±0.003	<u>0.208</u> ±0.091	0.727 ±0.005	0.735 ±0.003	<u>0.748</u> ±0.004
(DP)(39)(EnCTC)	<u>0.081</u> ±0.001	<u>0.641</u> ±0.006	<u>0.565</u> ±0.005	0.260 ±0.019	<u>0.582</u> ±0.004	<u>0.728</u> ±0.002	<u>0.217</u> ±0.086	0.726 ±0.003	0.736 ±0.002	<u>0.747</u> ±0.003
(DP)(39)(LP+EnCTC)	<u>0.081</u> ±0.001	<u>0.637</u> ±0.006	<u>0.562</u> ±0.005	0.262 ±0.023	<u>0.580</u> ±0.005	<u>0.726</u> ±0.002	<u>0.217</u> ±0.082	0.724 ±0.003	0.732 ±0.003	<u>0.745</u> ±0.003

Table 1: Evaluation results for the GOPT model. *CTC** denotes the standard CTC baseline. Results are reported as the mean and standard deviation ($\pm$) across five seeds. Bold and underlined text indicate performance exceeding the baseline, with bold representing the best result for each aspect. Models labeled (39) utilize the CMU 39-phone set, while (DP) refers to the proposed diphone units. (LP) and (EnCTC) represent label prior and conditional maximum entropy regularizations, respectively. Shaded rows distinguish models utilizing content-aware CTC.

3.4 Results

We compare our proposed context-dependent CTC methods with the TDNN-F (Kaldi) and standard CTC to evaluate their efficacy on Time Alignment (TA) accuracy and the resulting series of GOP features on the APA models. The TDNN-F baseline is well-known for its outstanding ability in audio-text alignment (Rousso et al., 2024), providing stable posteriors that serve as a robust benchmark for APA tasks. It outperforms all the other methods in the HierCB model, as shown in Table 2. But in GOPT shown in Table 1, WavLM(DP)(39) and WavLM(DP)(39) series outperform the TDNN-F (Kaldi) and standard CTC, and show the best results

in many phone- and word-aspects. This observation shows that the content-aware CTC can really help access pronunciation at fine-grained levels. HierCB relies on external hard alignments which may not benefit from the soft posterior density. GOPT is alignment-free and looks at the global feature sequence; it benefits more from the enriched acoustic information in our dense posteriors.

3.4.1 Phone-set Granularity: 39 Phones vs. Full Phoneme Set

We investigated whether the granularity of the units affects alignment and GOP quality. As shown in Figure 1, ASR models using the CMU 39-phone

ASR Models	Phoneme Score		Word Score (PCC)			Utterance Score (PCC)				
	MSE ↓	PCC ↑	Acc. ↑	Stress ↑	Total ↑	Acc. ↑	Comp. ↑	Fluency ↑	Prosodic ↑	Total ↑
TDNN-F(Kaldi) (Yan et al., 2025; Li et al., 2025)	0.078 ±0.001	0.660 ±0.002	0.608 ±0.011	0.385 ±0.045	0.625 ±0.011	0.757 ±0.006	0.685 ±0.153	0.835 ±0.004	0.826 ±0.003	0.787 ±0.004
WavLM(39)(CTC*) (Cao et al., 2024, 2026)	0.084 ±0.001	0.624 ±0.002	0.573 ±0.006	0.218 ±0.022	0.589 ±0.006	0.739 ±0.002	0.317 ±0.026	0.737 ±0.004	0.738 ±0.003	0.756 ±0.002
WavLM (DWFST-CTC)										
-	0.088 ±0.001	0.600 ±0.005	0.574 ±0.003	0.217 ±0.011	0.589 ±0.003	0.749 ±0.006	0.272 ±0.033	0.744 ±0.006	0.745 ±0.007	0.765 ±0.004
(LP)	0.114 ±0.002	0.432 ±0.007	0.430 ±0.006	0.097 ±0.008	0.442 ±0.006	0.635 ±0.006	0.099 ±0.016	0.665 ±0.011	0.653 ±0.009	0.645 ±0.008
(EnCTC)	0.088 ±0.002	0.602 ±0.006	0.566 ±0.008	0.203 ±0.020	0.581 ±0.009	0.743 ±0.006	0.248 ±0.046	0.738 ±0.005	0.740 ±0.005	0.762 ±0.004
(LP+EnCTC)	0.105 ±0.002	0.501 ±0.015	0.487 ±0.008	0.153 ±0.011	0.500 ±0.009	0.708 ±0.004	0.285 ±0.066	0.753 ±0.005	0.742 ±0.005	0.731 ±0.003
(39)	0.088 ±0.002	0.600 ±0.010	0.584 ±0.015	0.195 ±0.017	0.599 ±0.014	0.751 ±0.002	0.281 ±0.055	0.742 ±0.005	0.745 ±0.003	0.767 ±0.002
(39)(LP)	0.105 ±0.002	0.497 ±0.009	0.487 ±0.006	0.132 ±0.010	0.502 ±0.006	0.727 ±0.002	0.148 ±0.032	0.753 ±0.004	0.749 ±0.004	0.742 ±0.002
(39)(EnCTC)	0.104 ±0.001	0.494 ±0.010	0.490 ±0.008	0.143 ±0.011	0.505 ±0.007	0.727 ±0.003	0.185 ±0.035	0.749 ±0.004	0.741 ±0.004	0.743 ±0.003
(39)(LP+EnCTC)	0.087 ±0.002	0.611 ±0.007	0.570 ±0.012	0.267 ±0.006	0.586 ±0.011	0.742 ±0.008	0.427 ±0.049	0.773 ±0.004	0.773 ±0.005	0.761 ±0.007
(DP)	0.085 ±0.001	0.620 ±0.003	0.588 ±0.009	0.264 ±0.008	0.603 ±0.008	0.733 ±0.007	0.288 ±0.070	0.725 ±0.005	0.728 ±0.005	0.749 ±0.005
(DP)(LP)	0.107 ±0.002	0.491 ±0.012	0.466 ±0.014	0.153 ±0.013	0.479 ±0.012	0.708 ±0.007	0.306 ±0.047	0.743 ±0.005	0.737 ±0.007	0.727 ±0.006
(DP)(EnCTC)	0.084 ±0.001	0.623 ±0.002	0.584 ±0.002	0.254 ±0.006	0.598 ±0.002	0.741 ±0.004	0.383 ±0.031	0.730 ±0.005	0.734 ±0.005	0.754 ±0.003
(DP)(LP+EnCTC)	0.101 ±0.001	0.524 ±0.008	0.519 ±0.003	0.150 ±0.004	0.530 ±0.003	0.701 ±0.007	0.176 ±0.013	0.707 ±0.006	0.709 ±0.006	0.706 ±0.007
(DP)(39)	0.082 ±0.001	0.634 ±0.005	0.597 ±0.006	0.242 ±0.016	0.612 ±0.006	0.742 ±0.009	0.337 ±0.046	0.720 ±0.007	0.726 ±0.008	0.755 ±0.009
(DP)(39)(LP)	0.082 ±0.001	0.636 ±0.003	0.604 ±0.007	0.259 ±0.006	0.619 ±0.008	0.750 ±0.004	0.331 ±0.035	0.740 ±0.009	0.742 ±0.007	0.766 ±0.003
(DP)(39)(EnCTC)	0.083 ±0.002	0.632 ±0.004	0.590 ±0.010	0.232 ±0.016	0.605 ±0.009	0.748 ±0.006	0.261 ±0.033	0.737 ±0.009	0.743 ±0.010	0.764 ±0.008
(DP)(39)(LP+EnCTC)	0.084 ±0.001	0.628 ±0.003	0.593 ±0.009	0.231 ±0.012	0.608 ±0.008	0.736 ±0.006	0.287 ±0.039	0.725 ±0.015	0.730 ±0.014	0.751 ±0.008

Table 2: Evaluation results for HierCB (Yan et al., 2025) model.

set (pure phones) consistently outperformed their full-phoneme (stress-marked) counterparts across nearly all metrics. On TIMIT, WavLM(39)(LP) achieved a word TSE of 5.9% on the test set, compared to 6.6% for the standard WavLM(LP). This performance gain is likely due to the reduced confusion between acoustically similar phonemes, which simplifies the alignment task for the CTC loss.

3.4.2 Comparison of CTC Peak Mitigation Methods

We evaluated two primary methods for mitigating the peaky behavior and overconfidence in CTC models, LP and EnCTC, respectively. LP ensures the ASR model remains fair to rare phonetic symbols by normalizing the posteriors. Our results indicate that while standard WavLM-CTC lags behind the Kaldi baseline in word TSE (9.3% vs. 9.0%

on Buckeye), the integration of LP significantly closes this gap. As shown in Figure 1, WavLM(LP) achieves a word TSE of 7.3% on Buckeye, outperforming the Kaldi baseline by approximately 2% absolute. This suggests that accounting for the prior distribution of phones effectively mitigates the "lazy" alignment typical of standard CTC, leading to more precise boundary estimation.

Specifically, LP ensures the ASR model remains fair to rare phonetic symbols by normalizing the posteriors. This yields substantial gains in high-precision scenarios; for instance, the word ACC-50 on Buckeye increases from 44% (Standard CTC) to 69% with the addition of LP and phoneme level from 48% to 52%, shown in Table 3 and Table 4. In contrast, while EnCTC acts as a regularizer to ensure the ASR model remains cautious and avoids

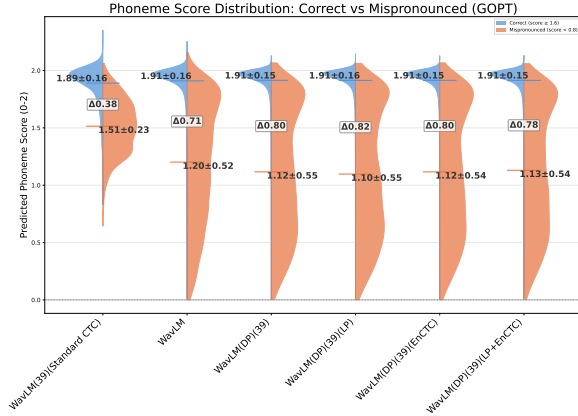


Figure 2: Distribution of GOPT phoneme scores for correct vs. mispronounced speech across various CTC-based ASR backbones.

overfitting to its own predictions, we observed that EnCTC is less effective than LP at reducing the BR.

Finally, we conducted experiments on the combination of both methods (LP+EnCTC). This hybrid approach yielded the best overall performance across all metrics. On the Buckeye dataset, WavLM(39)(LP+EnCTC) achieved the highest word ACC-50 score of 77.2%. These findings confirm that simultaneously addressing data bias (via LP) and model overconfidence (via EnCTC) is essential for achieving high-fidelity alignment suitable for downstream APA tasks.

3.4.3 DP vs. Regularizers

A notable observation is that LP still collapses on content-independent ASR models. Empirically, LP forces the model to predict rare phones more often. In a standard (monophone) CTC model, this might lead to insertion errors of rare phones into the blank spaces, destroying the alignment. However, in DP+39, the output space is more context-sensitive. The DP targets provide a stronger acoustic constraint that prevents LP from hallucinating rare symbols, instead using that prior to sharpen the transitions between phones.

3.4.4 Relational Studies: Alignment vs. APA Performance

We observed a clear correlation between TA accuracy and the PCC of the APA models. As shown in the comparison between Buckeye and TIMIT results, models with lower TSE (better alignment) consistently produced GOP features that led to higher word and utterance-level PCC in APA. For instance, the transition from standard WavLM-CTC

ASR Models	Buckeye / TIMIT dev / TIMIT test		
	P-TSE↓	P-ACC@25↑	P-ACC@50↑
TDNN-F(Kaldi)	26.3/5.2/6.1	26.1/34.4/32.5	63.6/80.9/76.5
WavLM(39)(CTC*)	<u>26.1/5.6/6.7</u>	<u>27.4/38.6/36.8</u>	54.9/73.2/68.8
WavLM (DWFST-CTC)			
-	26.3/7.1/8.2	21.0/25.9/24.9	48.9/59.2/56.0
(LP)	<u>26.1/7.2/8.4</u>	26.0/29.1/26.9	52.8/59.2/55.3
(EnCTC)	26.3/7.0/8.1	21.2/26.5/25.4	48.8/59.5/56.2
(LP+EnCTC)	<u>26.0/7.0/8.1</u>	<u>27.1/30.4/28.0</u>	54.2/61.0/56.3
(39)	<u>26.0/6.9/8.0</u>	23.5/27.1/26.2	51.1/61.1/57.6
(39)(LP)	<u>25.3/6.5/7.6</u>	<u>32.8/33.1/30.7</u>	59.0/64.9/60.1
(39)(EnCTC)	<u>26.2/6.6/7.6</u>	24.4/29.1/27.9	51.6/63.0/59.1
(39)(LP+EnCTC)	<u>25.2/6.4/7.5</u>	<u>32.9/33.8/31.8</u>	59.4/65.4/61.6
(DP)	38.2/6.8/6.9	9.7/22.4/23.5	26.9/57.4/57.9
(DP)(LP)	38.3/7.0/7.2	12.9/24.7/25.5	30.0/58.7/58.8
(DP)(EnCTC)	38.1/6.9/6.9	9.7/22.1/23.3	26.9/56.8/57.5
(DP)(LP+EnCTC)	38.5/7.0/7.1	9.5/22.3/22.7	26.6/56.5/57.2
(DP)(39)	38.2/6.8/7.0	9.9/22.1/23.3	27.1/57.4/58.0
(DP)(39)(LP)	38.2/6.8/7.0	9.8/22.6/23.5	27.0/57.5/58.1
(DP)(39)(EnCTC)	38.2/6.8/7.0	9.9/22.1/23.3	27.2/57.4/58.0
(DP)(39)(LP+EnCTC)	39.1/7.1/7.2	10.0/21.8/22.4	27.2/56.8/58.0

Table 3: A simplified result shows the TSE and ACC metrics on Buckeye corpus, TIMIT dev and test datasets, which only select the phoneme level (P-) and the tolerance at 25ms (@25) and 50ms (@50).

to WavLM(39)(LP+EnCTC) reduced phone TSE from 26.3% to 25.2% on Buckeye, which directly contributed to more reliable GOP feature extraction.

3.5 Scoring Results: Correct vs. Incorrect

We use the alignment-free APA model GOPT, which relies solely on CTC-GOP features, to test if our ASR changes actually help tell the difference between good and bad speech. As shown in Figure 2, the standard WavLM-CTC baseline has a scoring gap of 0.708. However, our best setup, WavLM(DP)(39)(LP), widens this gap to 0.816.

This 15% improvement in the scoring margin is a big deal for learners. While correct speech (score ≥ 1.6) gets high marks in all models, our new method is much better at catching mistakes. By fixing the peaky issue in CTC, the ASR model becomes more sensitive to the small acoustic errors that learners make. This makes the final scores more reliable and easier for a teacher to trust.

4 Discussion and Conclusion

Our results show that standard CTC’s “blank” dominance and peaky behavior are major barriers to stable pronunciation scoring. We demonstrate that context-aware CTC, which combines peaky mitigation and OCD targets, overcomes these limits. Our WavLM(DP)(39) series with LP or EnCTC has

ASR Models	Buckeye / TIMIT dev / TIMIT test		
	W-TSE↓	W-ACC@25↑	W-ACC@50↑
TDNN-F(Kaldi)	9.3/6.2/6.3	29.3/27.4/28.3	72.1/74.2/72.8
WavLM(39)(CTC*)	<u>9.0/7.9/7.8</u>	21.3/19.9/22.1	55.7/51.9/52.8
WavLM (DWFST-CTC)			
-	10.7/9.7/9.7	12.7/10.5/10.8	44.2/37.3/39.7
(LP)	<u>7.3/6.5/6.6</u>	<u>38.9/37.4/33.7</u>	69.8/61.8/60.0
(EnCTC)	10.7/9.7/9.7	12.7/11.1/11.0	43.9/37.3/39.8
(LP+EnCTC)	<u>7.3/6.4/6.7</u>	<u>38.2/37.3/32.3</u>	69.1/62.0/59.7
(39)	10.0/9.3/9.4	15.4/12.2/12.2	47.4/39.5/42.1
(39)(LP)	<u>6.0/5.9/6.0</u>	<u>46.8/41.4/38.3</u>	76.5/66.5/63.4
(39)(EnCTC)	9.9/9.1/9.0	16.5/13.0/14.0	48.4/41.4/43.2
(39)(LP+EnCTC)	5.9/5.6/5.8	47.7/43.2/40.1	77.2/67.4/65.7
(DP)	10.8/9.6/9.8	11.4/10.9/11.3	43.1/40.3/40.3
(DP)(LP)	<u>7.9/7.7/7.9</u>	<u>29.6/24.6/22.4</u>	63.9/54.7/54.8
(DP)(EnCTC)	10.9/9.7/9.8	11.3/10.8/11.2	42.9/40.0/39.9
(DP)(LP+EnCTC)	11.0/10.2/10.3	11.1/7.7/7.0	41.6/34.2/35.4
(DP)(39)	10.4/9.6/9.7	13.0/11.0/11.6	45.3/40.1/41.0
(DP)(39)(LP)	10.5/9.7/9.8	12.1/10.2/10.2	44.6/40.1/41.1
(DP)(39)(EnCTC)	10.4/9.6/9.7	13.0/11.0/11.6	45.3/40.2/41.0
(DP)(39)(LP+EnCTC)	10.6/9.8/10.0	12.7/9.0/9.0	44.2/36.7/37.5

Table 4: A simplified result shows the TSE and ACC on Buckeye corpus, TIMIT dev and test datasets, which only select the word level (W-) and the tolerance at 25ms (@25) and 50ms (@50).

now surpassed TDNN-F baseline. For APA, this soft and context-sensitive posterior provides a rich feature that captures the small acoustic details of a learner’s mistakes.

In conclusion, the context-aware CTC framework fixes standard CTC’s flaws for pronunciation tasks. We increased the scoring margin by 15% over the baseline. These findings prove that temporally dense and context-sensitive posteriors are essential for building reliable APA systems that provide stable, helpful feedback to learners.

5 Related Works

5.1 APA

APA systems evaluate second-language (L2) speech by providing fine-grained feedback across phoneme, word, and utterance levels (Gong et al., 2022; Yan et al., 2025). Most modern frameworks follow a two-stage pipeline: extracting segmental-level features and predicting scores through either unified (Gong et al., 2022; Chao et al., 2022) or hierarchical (Yan et al., 2025; Chao et al., 2023; Do et al., 2023) neural architectures. While unified models assess all granularities in a single decision, hierarchical models aggregate information from phoneme-level features up to the utterance level.

A fundamental building block in these models is the GOP feature, which measures the acoustic deviation between learner speech and native references

(Gong et al., 2022). While early systems relied on traditional ASR-derived GOP, current state-of-the-art models increasingly leverage self-supervised learning (SSL) backends, such as Wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and WavLM (Chen et al., 2022)—to extract robust contextual representations. These SSL models capture complex phonological patterns, significantly enhancing phoneme-level precision. In this work, we evaluate our proposed context-aware CTC-GOP features using both a unified architecture, GOPT (Gong et al., 2022), and a hierarchical model, HierCB (Yan et al., 2025).

Limitations

While the proposed method shows promise, several limitations remain. First, our evaluation is restricted to English as the target language with Mandarin L1 speakers. The efficacy of this approach for other language pairs and diverse L1 backgrounds requires further investigation.

Second, the transition from monophones to diphone units introduces a significant increase in the output layer dimensionality—from 39 to 39^2 possible combinations. This expansion increases computational overhead and may lead to data sparsity issues, as certain diphone transitions occur infrequently in training corpora.

Finally, our methodology is specifically designed to address the architectural constraints of CTC. Consequently, the adopted diphone-based context modeling may not be directly applicable or beneficial to non-CTC-based ASR frameworks, such as Transducer models.

Ethical Impact

We utilize publicly available and anonymized datasets (Librispeech, TIMIT, Buckeye, and speechocean762) for model training and evaluation. Our methodology focuses on improving speech technology for educational purposes and does not involve human subjects or the collection of private data. All authors comply with the ACL Code of Ethics and the relevant code of conduct for this venue.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. *wav2vec 2.0: A framework for self-supervised learning of speech representations*.

- Advances in Neural Information Processing Systems (NeurIPS)*, 33:12449–12460.
- Xinwei Cao, Zijian Fan, Torbjørn Svendsen, and Giampiero Salvi. 2024. [A framework for phoneme-level pronunciation assessment using CTC](#). In *Proc. of Interspeech*, pages 302–306.
- Xinwei Cao, Zijian Fan, Torbjørn Svendsen, and Giampiero Salvi. 2026. [Segmentation-free goodness of pronunciation](#). *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 34:796–807.
- Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2022. [3M: An effective multi-view, multi-granularity, and multi-aspect modeling approach to English pronunciation assessment](#). In *Proc. of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 575–582.
- Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2023. [A hierarchical context-aware modeling approach for multi-aspect and multi-granular pronunciation assessment](#). In *Proc. of Interspeech*, pages 974–978.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. [Wavlm: Large-scale self-supervised pre-training for full stack speech processing](#). *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 16(6):1505–1518.
- Xianzhao Chen, Yist Y. Lin, Kang Wang, Yi He, and Zejun Ma. 2023. [Improving frame-level classifier for word timings with non-peaky CTC in end-to-end automatic speech recognition](#). In *Proc. of Interspeech*, pages 2908–2912.
- Jan Chorowski, Adrian Łańcucki, Bartosz Kostka, and Michał Zpotoczny. 2019. [Towards using context-dependent symbols in CTC without state-tying decision trees](#). In *Proc. of Interspeech*, pages 4385–4389.
- Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. [Hierarchical pronunciation assessment with multi-aspect attention](#). In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- John S Garofolo, Lori F Lamel, William M Fisher, David S Pallett, Nancy L Dahlgren, Victor Zue, and Jonathan G Fiscus. 1993. [TIMIT acoustic-phonetic continuous speech corpus](#).
- Yuan Gong, Ziyi Chen, Iek-Heng Chu, Peng Chang, and James Glass. 2022. [Transformer-based multi-aspect multi-granularity non-native English speaker pronunciation assessment](#). In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7262–7266.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. [Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks](#). In *Proc. of the international conference on Machine learning (ICML)*, pages 369–376.
- Awni Hannun, Vineel Pratap, Jacob Kahn, and Wei-Ning Hsu. 2020. Differentiable weighted finite-state transducers. *arXiv preprint arXiv:2010.01003*.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhota, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. [Hubert: Self-supervised speech representation learning by masked prediction of hidden units](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 29:3451–3460.
- Wenping Hu, Yao Qian, Frank K. Soong, and Yong Wang. 2015. [Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers](#). *Speech Communication*, 67:154–166.
- Ruizhe Huang, Xiaohui Zhang, Zhaoheng Ni, Li Sun, Moto Hira, Jeff Hwang, Vimal Manohar, Vineel Pratap, Matthew Wiesner, Shinji Watanabe, Daniel Povey, and Sanjeev Khudanpur. 2024. [Less peaky and more accurate CTC forced alignment by label priors](#). In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11831–11835.
- Jiun-Ting Li, Bi-Cheng Yan, Yi-Cheng Wang, and Berlin Chen. 2025. [Multi-task pretraining for enhancing interpretable L2 pronunciation assessment](#). In *Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 531–536.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *Proc. of International Conference on Learning Representations (ICLR)*.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. [Librispeech: An asr corpus based on public domain audio books](#). In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.
- Aditya Kamlesh Parikh, Cristian Tejedor-Garcia, Ca-tia Cucchiari, and Helmer Strik. 2025. [Evaluating logit-based GOP scores for mispronunciation detection](#). In *Interspeech 2025*, pages 2405–2409.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). *Advances in neural information processing systems (NeurIPS)*, 32.

- Vijayaditya Pediti, Daniel Povey, and Sanjeev Khudanpur. 2015. [A time delay neural network architecture for efficient modeling of long temporal contexts](#). In *Proc. of Interspeech*, pages 3214–3218.
- Hao-Chen Pei, Hao Fang, Xin Luo, and Xin-Shun Xu. 2024. [Gradformer: A framework for multi-aspect multi-granularity pronunciation assessment](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 32:554–563.
- Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. [The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability](#). *Speech Communication*, 45(1):89–95.
- Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. [Multilayer perceptron and neural networks](#). *WSEAS Transactions on Circuits and Systems*, 8(7):579–588.
- Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. 2018. [Semi-orthogonal low-rank matrix factorization for deep neural networks](#). In *Proc. of Interspeech*, pages 3743–3747.
- Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. 2011. [The kaldı speech recognition toolkit](#). In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE Signal Processing Society.
- Rotem Rouso, Eyal Cohen, Joseph Keshet, and Eleanor Chodroff. 2024. [Tradition or innovation: A comparison of modern ASR methods for forced alignment](#). In *Proc. of Interspeech*, pages 1525–1529.
- Jiatong Shi, Nan Huo, and Qin Jin. 2020. [Context-aware goodness of pronunciation for computer-assisted pronunciation training](#). In *Proc. of Interspeech*, pages 3057–3061.
- S.M Witt and S.J Young. 2000. [Phone-level pronunciation scoring and assessment for interactive language learning](#). *Speech Communication*, 30(2):95–108.
- Bi-Cheng Yan, Jiun-Ting Li, Yi-Cheng Wang, Hsin Wei Wang, Tien-Hong Lo, Yung-Chang Hsu, Wei-Cheng Chao, and Berlin Chen. 2024. [An effective pronunciation assessment approach leveraging hierarchical transformers and pre-training strategies](#). In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1737–1747.
- Bi-Cheng Yan, Yi-Cheng Wang, Jiun-Ting Li, Meng-Shin Lin, Hsin-Wei Wang, Wei-Cheng Chao, and Berlin Chen. 2025. [ConPCO: Preserving phoneme characteristics for automatic pronunciation assessment leveraging contrastive ordinal regularization](#). In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zeyu Jin, Long Lin, and Daniel Povey. 2024. [Zipformer: A faster and better encoder for automatic speech recognition](#). In *Proc. of International Conference on Learning Representations (ICLR)*.
- Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2021. [Why does CTC result in peaky behavior?](#) *arXiv preprint arXiv:2105.14849*.
- Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yukai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. [speechocean762: An open-source non-native English speech corpus for pronunciation assessment](#). In *Proc. of Interspeech*, pages 3710–3714.
- Zeyu Zhao and Peter Bell. 2024. [Advancing CTC models for better speech alignment: A topological approach](#). In *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, pages 279–285.

A CTC-GOP implementation

A.1 Phone Inventory Reduction

When the acoustic model operates over a context-dependent phone set (e.g., with stress markers such as AA0, AA1, AA2), the log-probability matrix must be reduced to a pure-phone inventory before GOP computation. This mirrors the `ComputeLpps()` procedure in the speechocean762 official implementation² (Zhang et al., 2021).

Given a log-probability matrix $M \in \mathbb{R}^{T \times C}$, where T is the number of frames and C is the full phone inventory size, and a surjective mapping $\phi : \{0, \dots, C-1\} \rightarrow \{0, \dots, K-1\}$ from tokens to pure phones, we compute the reduced log-probability matrix $M' \in \mathbb{R}^{T \times K}$ as:

$$M'_{t,k} = \log \sum_{j: \phi(j)=k} \exp(M_{t,j}) \quad (12)$$

To ensure numerical stability, we apply the grouped log-sum-exp (log-space marginalization) trick. For each frame t and pure phone k , let $m_{t,k} = \max_{j: \phi(j)=k} M_{t,j}$. Then:

$$M'_{t,k} = m_{t,k} + \log \sum_{j: \phi(j)=k} \exp(M_{t,j} - m_{t,k}) \quad (13)$$

If no token maps to a particular pure phone k at frame t (i.e., the set $\{j : \phi(j) = k\}$ is empty), we define $M'_{t,k} = -\infty$. This reduction is implemented efficiently using `scatter_reduce` with the `amax` operator for computing $m_{t,k}$, and `scatter_add` for the shifted exponential summation, avoiding explicit loops over the vocabulary.

²<https://github.com/kaldi-asr/kaldi/blob/master/src/bin/compute-gop.cc>

A.2 GOP Feature Computation

Let $Y_c = (s_1, s_2, \dots, s_N)$ denote the canonical phone sequence for a given utterance, where N is the number of phones. We define the *LPP* of the canonical sequence as:

$$lpp = -\mathcal{L}_{\text{CTC}}(X, Y_c) \quad (14)$$

where $\mathcal{L}_{\text{CTC}}$ denotes the CTC loss (i.e., the negative log-likelihood under the CTC formulation). It is notable that the input posterior to the CTC loss could select log posterior rather than pure probability one (Cao et al., 2026) to avoid numerical underflow errors when multiplying probabilities over longer speech segments.

For each phone position $i \in \{1, \dots, N\}$, we construct a set of hypothesis sequences through two operations:

- **Deletion:** Remove the phone at position i , yielding $Y_c^{(i, \text{del})} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_N)$.
- **Substitution:** Replace the phone at position i with every phone $q \in \{1, \dots, K-1\}$, yielding $Y_c^{(i, q)} = (s_1, \dots, s_{i-1}, q, s_{i+1}, \dots, s_N)$.

This produces $N \times K$ hypothesis sequences in total (one deletion and $K-1$ substitutions per position). All hypothesis sequences are batched into a single padded tensor and evaluated simultaneously via PyTorch(Paszke et al., 2019)’s `F.ctc_loss` with `reduction='none'`, enabling efficient GPU-parallel computation.

The **LPR** for position i and hypothesis q is defined as:

$$\text{LPR}_{i,q} = \mathcal{L}_{\text{CTC}}(X, s^{(i,q)}) - lpp \quad (15)$$

This quantity measures how much less likely the modified sequence is compared to the canonical pronunciation. A positive **LPR** for the substitution with the canonical phone at position i indicates correct pronunciation; values near zero or negative suggest mispronunciation.

A.3 Feature Representation

For each phone position i , the extracted feature vector is:

$$\mathbf{f}_i = [\phi(s_i), lpp, \text{LPR}_{i,0}, \text{LPR}_{i,1}, \dots, \text{LPR}_{i,K-1}] \quad (16)$$

where $\phi(s_i)$ is the pure phone identity and $\text{LPR}_{i,0}$ corresponds to the deletion hypothesis. The resulting feature matrix $\mathbf{F} \in \mathbb{R}^{N \times (K+2)}$ is stored per utterance for downstream mispronunciation detection or scoring.

A.4 CTC Forward Algorithm

For reference, the CTC forward log-probability is computed via the standard dynamic programming recursion. Let $\mathbf{s}' = (\emptyset, s_1, \emptyset, s_2, \emptyset, \dots, \emptyset, s_N, \emptyset)$ be the blank-augmented label sequence of length $L = 2N + 1$. The forward variable $\alpha_{l,t}$ represents the log-probability of emitting the partial sequence $\mathbf{s}'_{1:l}$ over frames 1 through t :

$$\alpha_{l,t} = \log p(\mathbf{s}'_{1:l}, X_{1:t}) \quad (17)$$

Let $\Psi(t-1, \mathcal{L}) = \log \sum_{l' \in \mathcal{L}} \exp(\alpha_{l',t-1})$. The recursion proceeds as:

$$\alpha_{l,t} = \begin{cases} \Psi(t-1, \mathcal{A}(l)) + \log p(\mathbf{s}'_l | t), & \text{if } l \bmod 2 = 0 \\ \Psi(t-1, \mathcal{B}(l)) + \log p(s_{\lfloor l/2 \rfloor} | t), & \text{otherwise} \end{cases}$$

where $\mathcal{A}(l) = \{l, l-1\}$ for blank positions, and $\mathcal{B}(l) = \{l, l-1\}$ when $s_{\lfloor l/2 \rfloor} = s_{\lfloor l/2 \rfloor - 1}$ (repeated label), or $\{l, l-1, l-2\}$ otherwise. The total sequence log-probability is obtained as:

$$\log p(\mathbf{s} | X) = \text{logaddexp}(\alpha_{L,T}, \alpha_{L-1,T}) \quad (18)$$

A.5 Implementation Details

The pipeline operates as follows: (a) a pre-trained CTC acoustic model produces frame-level log-probabilities from audio; (b) canonical phone sequences are derived either from a lexicon-based word-to-phone expansion or from an explicit phoneme-level transcript; (c) GOP features are computed per utterance and saved as NumPy arrays. The batched CTC evaluation avoids the $\mathcal{O}(N \times K)$ sequential forward passes that would be required by a naïve implementation, yielding substantial speedup on GPU hardware.

B ASR and APA models training configures

In this work, we conduct experiments to train the ASR models on the Librispeech (Panayotov et al., 2015) corpus, a 960-hour training set. We utilize `torchaudio.bundle` to implement the backbone,

selecting WAVLM_LARGE as its initialized model parameters³. For some control groups, such as WavLM(39) and WavLM(DP)(39), they are based on WavLM and WavLM(DP), respectively, and further finetuned on the train-clean 100 hours training subset. In detail, we freeze all the WavLM encoder layers, and finetune only the output layers. For the DP ones, the output layer was designed with a single feedforward layer following two CD embedding layers (Chorowski et al., 2019)⁴. The feature extractor component of the WavLM model remains unchanged. Our models use phonemes as the modeling units. For the standard CTC, we adopt the implementation of Cao et al. (2024)⁵, but change the backbone SSL model from wav2vec2.0 to WavLM. For the ASR model optimization, we employ the AdamW (Loshchilov and Hutter, 2019) optimizer, with a learning rate initialized from 10^{-4} .

Throughout our experimental setup, we utilize Zhao and Bell (2024)⁶ project to conduct ASR experiments, in which Kaldi (Povey et al., 2011) for data preparation and PyTorch (Paszke et al., 2019) for neural network training, with k2-fsa⁷ serving as the backend for the DWFST framework. The computation of CTC-GOP features follows the idea in Cao et al. (2024) but is further optimized for its stability and the running speed.

For the comparative methods, LP and EnCTC, we adopt the implementations^{8 9}, adjusting them to fit in the implementation of computing loss in the k2-fsa (Yao et al., 2024)¹⁰ framework. For LP, we change to a smart strategy, accumulating its label prior to each mini-batch to achieve a gradual adjustment. The a for the label prior weight is set to 0.2, while the b for the conditional maximum entropy weight is set to 0.1.

For GOPT and HierCB models training, we adhere to the hyperparameters from Gong et al. (2022); Yan et al. (2025) for training of our APA model. To quantify epistemic uncertainty, we train

with five different random seeds.

³https://docs.pytorch.org/audio/stable/generated/torchaudio.pipelines.WAVLM_LARGE.html

⁴https://github.com/chorowski-lab/pytorch-asr/blob/master/att_speech/modules/decoders/advanced_decoder.py

⁵<https://github.com/frank613/CTC-based-GOP>

⁶<https://github.com/ZhaoZeyu1995/BenNevis>

⁷<https://github.com/k2-fsa/k2>

⁸https://github.com/huangruizhe/audio/blob/aligner_label_priors/examples/asr/librispeech_alignment/loss.py

⁹https://github.com/liuhu-bigeye/enctc.crnn/blob/master/pytorch_ctc/ctc_ent.py

¹⁰<https://github.com/k2-fsa/k2>

A Survey of Automated Presentation Coaching: Systems, Methods, and Open Challenges

Wen Liang^{1,2}, Li Siyan¹, Zackary Rackauckas³, Julia Hirschberg¹

¹Columbia University, USA ²Red Hat, USA ³RoleGaku, USA

{wl2904, siyan.li, zcr2105}@columbia.edu, julia@cs.columbia.edu

Abstract

Automated coaching for oral presentations sits at the intersection of computer-assisted pronunciation training (CAPT), prosody modeling, and speech synthesis, yet no prior work has systematically surveyed and compared existing systems along these dimensions. This survey reviews and categorizes automated presentation coaching systems, spanning pronunciation tutors, fluency and prosody coaches, multimodal trainers, and conference Q&A practice tools. We introduce a five-dimensional task taxonomy—covering segmental pronunciation, lexical stress, suprasegmental prosody, pacing, and content faithfulness—and explicitly map surveyed systems onto it to reveal coverage gaps. We further review the core technical methods these systems employ: TTS-based exemplar generation and diagnostic methods for pronunciation, prosody, and fluency assessment. Key open challenges include the scarcity of annotated presentation corpora, achieving accent-fair feedback across diverse L1 backgrounds, and delivering low-latency diagnostics for real-time rehearsal.

1 Introduction

Oral presentations in English, including technical talks, research seminars, and product demos, impose demands well beyond everyday conversation. Presenters must articulate specialized terminology accurately, manage pacing relative to slide transitions, and modulate prosody so that discourse structure is clear to listeners. For English as a Second Language (ESL) speakers, these demands are compounded by the tendency to carry over sound patterns from one’s native language into English, accent-specific prosodic patterns, and limited opportunities for realistic rehearsal (Munro and Derwing, 1995; Derwing and Munro, 2005).

The last two decades have seen growing interest in *automated coaching* for oral presentations,

spurred by progress in computer-assisted pronunciation training (CAPT) (Witt and Young, 2000; Cucchiaroni et al., 2009), neural text-to-speech (TTS) (Chen et al., 2024; Ren et al., 2021), and multimodal analysis (Baltrušaitis et al., 2019). Systems such as Rhema (Tanveer et al., 2015), Mirror Mirror (Schneider et al., 2015), and recent LLM-augmented Q&A coaches (Aiba et al., 2024) each target different facets of presentation skill. Yet the field currently lacks a unified survey that (a) catalogs and compares these systems, (b) identifies which presentation dimensions they address, and (c) exposes the problems that remain.

This paper fills that gap. We survey automated coaching systems for oral presentations, focusing on speech-based dimensions: pronunciation, lexical stress, prosody, pacing, and content faithfulness. Our scope spans both L2-specific and general public-speaking tools, since many techniques transfer across these settings. We deliberately set aside visual and gesture coaching (Schneider et al., 2015; Damian et al., 2015) except when they are integrated with speech feedback, and we do not aim to cover the full literature on general-purpose ASR or spoken language assessment.

Contributions. Concretely, this survey: (1) introduces a five-dimensional task taxonomy for presentation coaching and maps surveyed systems to reveal coverage gaps; (2) systematically reviews and categorizes existing automated presentation coaching systems; (3) reviews the core technical methods – TTS-based exemplar generation and diagnostic approaches – that underpin these systems; and (4) discusses open challenges in corpora, accent fairness, real-time deployment, and the gap between research and industry practice.

Literature Search Strategy. To ensure comprehensive coverage, we conducted a systematic search across ACL Anthology, IEEE Xplore, ISCA Archive, Google Scholar, and Semantic Scholar using queries combining terms such as “presenta-

tion coaching,” “pronunciation training,” “CAPT,” “prosody assessment,” “speech fluency,” “TTS coaching,” and “L2 speaking.” We included peer-reviewed publications from 1997 to 2025 that directly address automated coaching or assessment of oral presentation or spoken language skills. We supplemented keyword-based search with citations from key papers (Witt and Young, 2000; Golonka et al., 2014; Aiba et al., 2024), following established survey methodology (Frederick Eneye et al., 2025). Studies focused exclusively on visual or gestural coaching without a speech component were excluded, as were general-purpose ASR or TTS papers. This process yielded 15 representative systems and approximately 50 supporting references spanning CAPT, prosody modeling, TTS, and educational technology.

Organization. Section 2 reviews foundational technologies. Section 3 presents the five-dimensional task taxonomy that organizes the remainder of the paper, along with the system inputs, outputs, and operating modes that connect the taxonomy to practical coaching workflows. Section 4 surveys existing systems and maps them onto the taxonomy. Section 5 reviews the core technical methods these systems employ. Section 6 covers datasets and evaluation. Section 7 discusses remaining open problems and future directions, including the gap between research and industry deployment. Section 8 concludes.

2 Background: Foundational Technologies

We briefly review the core technologies that underpin automated presentation coaching: pronunciation assessment, prosody analysis, shadowing pedagogy, and neural TTS. See detailed discussions of diagnostic and synthesis methods in Section 5.

Computer-Assisted Pronunciation Training (CAPT). CAPT systems localize segmental errors using Goodness of Pronunciation (GOP) scores (Witt and Young, 2000) or ASR confidence measures (Cucchiarini et al., 2009). Earlier approaches relied on HMM forced alignment (Rabiner, 1989; Franco et al., 1997); modern systems leverage CTC (Graves et al., 2006; Cao et al., 2024) and self-supervised representations (Baevski et al., 2020; Hsu et al., 2021) for alignment-free mispronunciation detection (Xu et al., 2021; Gong et al., 2022). These methods form the diagnostic backbone of most pronunciation coaching systems.

Prosody Analysis. Effective presentations utilize prosody (intonation, phrasing, rhythm) to signal structure (Hirschberg, 2004). Listener-impression studies (Shoda et al., 2023) confirm that even small pitch and timing adjustments significantly affect perceived speaker competence. Assessment typically compares log- F_0 contours and duration patterns between learner and reference utterances (Sakoe and Chiba, 1978; Rosenberg, 2010).

Shadowing Pedagogy. Exemplar-guided shadowing (imitation) is well established for improving L2 pronunciation, prosody, and fluency (Hori, 2008; Kadota, 2019; Hamada, 2018). When mimicking short auditory demonstrations with explicit focus on speech rate and prominence, learners experience improvements in timing, stress, and intonation (Hsieh et al., 2013). This pedagogy directly motivates the use of TTS to generate controllable, adjustable exemplars for coaching at scale.

Neural Text-to-Speech (TTS). Recent non-autoregressive flow-matching TTS models (e.g., F5-TTS (Chen et al., 2024), Voicebox (Le et al., 2023), CosyVoice 2 (Du et al., 2024)) synthesize highly natural speech with real-time factors below 1. These systems offer fine-grained control over speaking rate, pause insertion, and emphasis, making them well-suited for generating coaching exemplars. Zero-shot style transfer from short enrollment clips enables personalized references (Jia et al., 2018; Casanova et al., 2022).

3 Presentation Coaching Taxonomy

Building on the foundational technologies reviewed above, we formalize a five-dimensional taxonomy for automated presentation coaching. The taxonomy organizes presentation skills by the nature of the feedback they require and the methods available to assess them, providing a systematic framework for comparing existing systems and identifying coverage gaps. Figure 1 illustrates the structure.

3.1 Taxonomy Dimensions

We define each dimension, its assessment methods, and the kind of feedback it yields; Section 4 maps the full set of surveyed systems onto these dimensions, and Table 1 provides the complete mapping.

Pronunciation (segmental) covers phone- and word-level correctness, including salient vowel/-consonant contrasts and technical terminology. Assessment relies on GOP (Witt and Young, 2000),

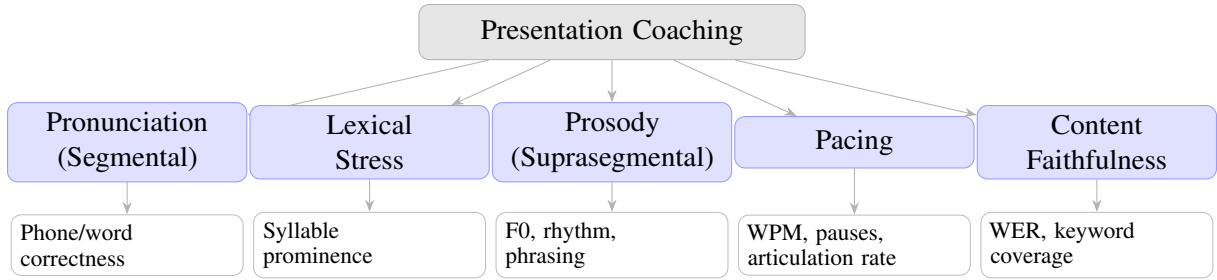


Figure 1: Five-dimensional taxonomy for automated presentation coaching. Existing systems (Table 1) cover different subsets; lexical stress and content faithfulness are the least-addressed dimensions.

CTC-based (Cao et al., 2024), or self-supervised methods (Xu et al., 2021); feedback for this dimension identifies top- k error phones. This is the most widely addressed dimension in the literature.

Lexical stress addresses syllable prominence in multi-syllabic words (e.g., *AL-go-rithm*). Feedback identifies the incorrect stress position and proposes re-stress drills. Despite its documented impact on comprehensibility (Munro and Derwing, 1995), this dimension is almost entirely neglected by existing systems (Korzekwa et al., 2021).

Prosody (suprasegmental) includes intonation (F0), phrasing, rhythm, and intensity contours that signal discourse structure. Feedback often includes F0 RMSE and Pearson r vs. a reference. Several systems address prosody with varying granularity.

Pacing covers words-per-minute (WPM), articulation rate, and pause placement relative to punctuation and slide boundaries. Feedback highlights WPM deviation and pause statistics.

Content faithfulness measures coverage of key content without insertions or omissions, flagging missing keywords per slide. Providing the ASR with a list of expected technical terms from the slides helps it correctly transcribe domain-specific vocabulary that a general-purpose model might otherwise miss. This is the least-addressed dimension in the literature.

3.2 Inputs, Outputs, and Assumptions

Beyond the five coaching dimensions, systems also differ in their input requirements and output granularity, with each input-output pattern tied to specific taxonomy dimensions. On the **input** side, systems targeting the *pronunciation* and *lexical stress* dimensions require, at minimum, a text transcript and a learner recording for forced alignment (Witt and Young, 2000; Neri et al., 2002; Strik et al., 2009; Xu et al., 2021). Systems addressing *prosody* and *pacing* additionally assume slide-aligned scripts

with section boundaries to enable discourse-level analysis (Hincks, 2005; Chen et al., 2014; Schneider et al., 2015). Systems targeting *content faithfulness* require the reference slide content for keyword matching. Systems that support personalized exemplars across any dimension may further accept a brief enrollment clip for voice cloning (Jia et al., 2018; Casanova et al., 2022).

On the **output** side, feedback granularity maps directly to the taxonomy. *Pronunciation* and *lexical stress* systems produce per-word or per-phone error flags (Witt and Young, 2000; Cao et al., 2024; Korzekwa et al., 2021). *Prosody* and *pacing* systems report global scores such as F0 deviation, articulation rate, and pause frequency (Shen et al., 2021; Saito et al., 2023). *Content faithfulness* systems flag missing keywords and WER per slide (Aiba et al., 2024). Multimodal trainers additionally output behavioral feedback on gaze and gesture (Schneider et al., 2015; Damian et al., 2015; Ramanarayanan et al., 2015). Across these systems, text normalization, grapheme-to-phoneme conversion, and forced or CTC-based alignment are standard preprocessing steps, while phonetic labels from the learner are generally not required.

3.3 System Modes

Given these inputs and outputs, coaching systems operate in three complementary modes, each serving different taxonomy dimensions. **Exemplar generation** (Section 5) produces controllable target reads per slide and for glossary items, directly supporting the *pronunciation*, *lexical stress*, and *prosody* dimensions by providing reference audio that models correct segment production, stress patterns, and intonation contours. **Comparative assessment** (Section 5) aligns learner audio with a reference to localize deviations, enabling diagnostics across all five taxonomy dimensions—from phone-level pronunciation errors to slide-level con-

tent omissions. A **mixed-initiative practice loop** turns diagnoses into focused drills targeting the weakest dimensions (e.g., re-stress a syllable for *lexical stress*; maintain a pre-equation pause for *pacing*), allowing the learner to re-record and track progress. This loop is implicit in several prior systems (Shen et al., 2021; Saito et al., 2023; Aiba et al., 2024) but rarely articulated explicitly for slide-based presentations. The next section surveys existing systems and maps them onto the taxonomy dimensions defined above.

4 Survey of Existing Presentation Coaching Systems

With the taxonomy in place, we now survey automated systems that target presentation or spoken English coaching. We organize them into four categories: CAPT-based pronunciation systems, prosody and fluency coaches, multimodal trainers, and Q&A/interaction coaches. Table 1 maps each system onto the five taxonomy dimensions.

4.1 CAPT-Based Pronunciation Systems

Early pronunciation coaching focused on segmental accuracy. Franco et al. (1997) pioneered automatic pronunciation scoring for language instruction using HMM-based forced alignment and log-likelihood scores, establishing the feasibility of system-generated feedback on individual phones. Neri et al. (2002) examined the pedagogy-technology interface in CAPT systems, showing that explicit error localization and corrective feedback improve learner outcomes beyond passive listening. Strik et al. (2009) compared GOP, posterior-based, and ASR-confidence approaches for automatic pronunciation error detection, finding that methods vary considerably across L1 backgrounds, motivating per-cohort calibration. More recently, Xu et al. (2021) demonstrated that wav2vec 2.0 representations substantially outperform HMM-based systems on L2 English mispronunciation detection benchmarks. Korzekwa et al. (2021) specifically addressed lexical stress detection in L2 English, using data augmentation and attention-based models to identify incorrect primary-stress placement in multi-syllabic words.

However, these systems share a common limitation: they operate on isolated words or short, read sentences, not on the semi-scripted, long-form speech characteristic of presentations. Feedback is typically provided post-hoc as a list of error flags,

with no practice loop or exemplar generation.

4.2 Fluency and Prosody Coaching

Moving beyond segmental accuracy, a second line of work targets fluency and suprasegmental quality. Hincks (2005) developed one of the earliest computer-aided systems specifically targeting spoken English for academic presentations, providing visual feedback on speaking rate and pitch variation. Shen et al. (2021) presented an interpretable model predicting L2 fluency from acoustic features including articulation rate, pause frequency, and phonation time, identifying the features that most strongly correlate with human ratings across diverse L1 backgrounds. Saito et al. (2023) conducted a comprehensive program on automated L2 comprehensibility assessment, arguing that comprehensibility, rather than narrow segmental accuracy, is a more robust target for coaching.

On the synthesis side, prior work (Onda et al., 2024, 2025) showed that generative spoken language models can simulate and modulate foreign-accented prosody, enabling the generation of corrected exemplars that retain the learner’s vocal identity while gaining similarity to target prosody patterns.

Zechner et al. (2009) developed automated scoring of non-native spontaneous speech for high-stakes testing (e-rater), covering fluency, pronunciation, and prosody holistically.

4.3 Multimodal Presentation Trainers

A third wave of systems extends speech coaching to the full presentation context, incorporating visual and behavioral signals together with audio. Schneider et al. (2015) developed *Mirror Mirror*, a multimodal presentation coaching system that provides automated feedback on body language, eye contact, and filler words in addition to speech rate. Damian et al. (2015) presented an intelligent tutoring system that analyzes posture, gesture, and vocal quality to generate structured coaching advice for novice presenters, grounded in public speaking pedagogy (Lucas, 2014). Ramanarayanan et al. (2015) assessed multimodal communication skills—including speech, gaze, and gesture—using behavioral signals extracted automatically, demonstrating feasibility for scalable deployment.

While the above systems combine multiple modalities, other work targets individual speech dimensions more directly. Chen et al. (2014) evaluated presentation skills in a tutoring context using

speech recognition and audience feedback. These systems demonstrate the feasibility of presentation-specific feedback loops, though they typically address only one or two of the dimensions in our defined taxonomy at a time.

4.4 Q&A and Interaction-Based Coaching

The most recent systems leverage large language models to extend coaching to the interactive components of presentations. Aiba et al. (2024) proposed a multimodal system for conference Q&A practice, combining ASR with ChatGPT-based question generation and TTS to simulate domain-relevant follow-up questions. Learners deliver a short talk; the system transcribes it, generates content-contingent questions, and reads them aloud, enabling repeated practice of spontaneous responses. Their user study reported reducing presentation anxiety and improving perceived preparedness among first-time conference participants.

Most recently, Chen et al. (2025) introduced PresentCoach, a dual-agent system in which one agent generates benchmark presentation videos from slides using personalized voice synthesis, while another evaluates the learner’s recording against these exemplars and delivers structured feedback. It is the first system to integrate slide-aware exemplar generation with multi-dimensional assessment in a single LLM-driven pipeline, covering four of the five taxonomy dimensions (pronunciation, prosody, pacing, and content faithfulness). Neural phonetic posteriorgrams (PPGs) (Churchwell et al., 2024) offer a complementary phone-level diagnostic suitable for such interactive loops.

4.5 Comparison and Discussion

Together, these four system categories cover a wide arc of presentation skills, yet no single system spans all. Table 1 compares the surveyed systems along the five coaching dimensions from our taxonomy (Section 3), whether the system provides real-time feedback, and whether it targets L2 speakers specifically.

Several patterns emerge from Table 1. First, **lexical stress is almost entirely neglected**: only Korzekwa et al. (2021) directly targets stress detection, despite its well-documented importance for L2 comprehensibility (Munro and Derwing, 1995). Second, **real-time feedback remains rare**: only two surveyed systems provide in-session feedback (Schneider et al., 2015; Aiba et al., 2024); most systems assess post-hoc. Third, **content faith-**

fulness and pacing are often decoupled: systems either check what was said (ASR-based) or how fast (WPM), but seldom do both with slide-aligned structure. Fourth, **no existing system covers all five dimensions simultaneously**: the most recent system, PresentCoach (Chen et al., 2025), covers four (pronunciation, prosody, pacing, and content) but still omits lexical stress, illustrating that integrated, multi-dimensional coaching remains an open challenge. These gaps are compounded by limitations in available data: most CAPT corpora (Zhao et al., 2018; Zhang et al., 2021; Garofolo et al., 1993) focus on isolated words or short sentences, lacking slide structure, domain terminology, and discourse-level features, while large-scale corpora such as LibriSpeech (Panayotov et al., 2015), Common Voice (Ardila et al., 2020), and TED-LIUM (Hernandez et al., 2018) are not designed for L2 coaching. In summary, existing systems address individual dimensions but do not combine exemplar-based practice with fine-grained diagnostics in a unified, presentation-specific workflow (Golonka et al., 2014).

5 Coaching Systems Methods

The systems surveyed in Section 4 draw on two complementary families of methods: **TTS-based exemplar generation**, which produces reference targets for learner practice, and **diagnostic techniques**, which assess how far a learner deviates from those targets. This section reviews both families of approaches employed by prior systems.

5.1 TTS-Based Exemplar Generation

Several surveyed systems use synthesized or recorded model speech as a reference for learner practice (Hincks, 2005; Schneider et al., 2015; Aiba et al., 2024). We review the TTS capabilities these systems exploit and the workflows they follow.

5.1.1 TTS Capabilities for Coaching

Modern neural TTS offers three capabilities that the surveyed systems leverage: **controllability** over rate, pauses, and emphasis; **code-switching** support for technical terms; and **zero-shot voice transfer** from short enrollment clips. Because recent flow-matching models (Chen et al., 2024; Le et al., 2023) and streaming architectures such as CosyVoice 2 (Du et al., 2024) achieve sub-second latency, exemplars can now be rendered on-the-fly during rehearsal. Two controls are most relevant:

System	Pron.	Stress	Prosody	Pacing	Content	Real-time	L2-specific
Franco et al. (1997)	✓						✓
Neri et al. (2002)	✓						✓
Hincks (2005)			✓	✓			✓
Strik et al. (2009)	✓						✓
Zechner et al. (2009)	✓		✓	✓			✓
Chen et al. (2014)	✓		✓	✓	✓		
Damian et al. (2015)			✓	✓			
Schneider et al. (2015)				✓		✓	
Ramanarayanan et al. (2015)	✓		✓				
Shen et al. (2021)	✓		✓	✓			✓
Korzekwa et al. (2021)		✓					✓
Xu et al. (2021)	✓						✓
Saito et al. (2023)	✓		✓				✓
Aiba et al. (2024)					✓	✓	✓
Chen et al. (2025)	✓		✓	✓	✓		

Table 1: Comparison of existing presentation and spoken-language coaching systems across five taxonomy dimensions: “Pron.” = segmental pronunciation; “Stress” = lexical stress; “Content” = faithfulness/content coverage; “Real-time” = feedback provided during or immediately after recording. Checkmarks indicate that the system explicitly addresses the dimension.

tempo bands that constrain WPM, and *emphasis marks* that elicit pitch accents on key terms.

5.1.2 Workflow and Pedagogical Strategies

Generalizing patterns from prior systems (Hincks, 2005; Schneider et al., 2015) and grounded in shadowing pedagogy (Kadota, 2019; Hamada, 2018), a typical TTS-based coaching pipeline proceeds as follows: (1) render an *anchor* exemplar at a conservative tempo (120–140 WPM) and an optional *target* at a faster tempo (150–170 WPM); (2) record the learner in short chunks (5–12 s per section); (3) align learner audio with the reference using CTC (Graves et al., 2006) or DTW (Sakoe and Chiba, 1978) and compute deviations; (4) surface focused drills based on the highest-error dimensions. Across these systems, effective exemplars tend to be short (<12 s) to prevent cognitive overload and offer multiple speed bands and vocabulary difficulty gradations to accommodate different proficiency levels (Schneider et al., 2015; Golonka et al., 2014). Known limitations include over-constraining style when exemplars are treated as a single correct read, and synthetic emphasis that may not match domain conventions.

5.2 Pronunciation and Fluency Diagnostics

While TTS provides the coaching *targets*, diagnostic methods provide the *assessments* (Witt, 2012; Jurafsky and Martin, 2026). The surveyed systems employ three families of diagnostic techniques,

each addressing different taxonomy dimensions.

5.2.1 GOP and CTC-Based Assessment

Goodness of Pronunciation (GOP) (Witt and Young, 2000), used by several surveyed systems (Franco et al., 1997; Zechner et al., 2009; Strik et al., 2009), scores each phone by comparing how strongly the acoustic evidence supports the intended sound vs. the most likely alternative, averaged across the phone’s duration. A high GOP score indicates correct pronunciation; a low score flags a likely mispronunciation. Modern “segmentation-free” variants (Cao et al., 2024) derive boundaries from CTC posteriors (Graves et al., 2006) rather than forced alignment, improving robustness to disfluent L2 speech. Self-supervised models (wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), WavLM (Chen et al., 2022)) further advance this line (Xu et al., 2021; Gong et al., 2022), and neural PPGs (Churchwell et al., 2024) provide an alternative backbone for fine-grained phone-level scoring. Thresholds are calibrated per L1 on small rated sets, optimizing equal error rate or UAR against human annotations.

5.2.2 Clone-and-Compare (Personalized Reference)

To control for timbre and speaker style, several systems (Onda et al., 2024, 2025) synthesize an idealized rendition in the learner’s voice via voice conversion (Qian et al., 2019) or zero-shot TTS (Jia

Corpus	Hrs	L2	Long	Accent	Prosody
TIMIT (Garofolo et al., 1993)	5				
L2-ARCTIC (Zhao et al., 2018)	26	✓		✓	
Speechocan762 (Zhang et al., 2021)	70	✓			✓
Speech Accent Archive (Weinberger, 2015)	–	✓		✓	
LibriSpeech (Panayotov et al., 2015)	960				
Common Voice (Ardila et al., 2020)	1400+			✓	
TED-LIUM 3 (Hernandez et al., 2018)	452		✓		✓
GigaSpeech (Chen et al., 2021)	10k		✓		
EpaDB (Vidal et al., 2019)	3	✓			✓
Speak & Improve (Knill et al., 2024)	340	✓		✓	

Table 2: Major corpora used in presentation coaching research. “Long” = contains long-form or discourse-level speech; “Accent” = includes L1 or accent labels; “Prosody” = includes prosodic annotations. No existing corpus satisfies all four criteria simultaneously.

et al., 2018; Casanova et al., 2022), then compute MFCC/SSL (Davis and Mermelstein, 1980; Baevski et al., 2020) distance curves aligned with monotonic DTW (Sakoe and Chiba, 1978). Distance peaks flag likely mispronunciations while controlling for speaker-specific timbre. End-to-end ASR (Radford et al., 2023; Gulati et al., 2020) also yields word durations and confidence posteriors as complementary signals.

5.2.3 Prosody, Pacing, and Faithfulness

Prosodic quality is computed on $\log-F_0$ after voicing decisions (YAAPT (Zahorian and Hu, 2008)): per-slide RMSE and Pearson r capture intonation match, while word- or phrase-level duration RMSE captures rhythmic alignment. These suprasegmental metrics are used by several surveyed systems (Hincks, 2005; Shen et al., 2021; Saito et al., 2023). Pacing metrics (WPM deviation, articulation rate, pause precision/recall) appear in systems targeting real-time feedback (Schneider et al., 2015). For content faithfulness, constrained-LM ASR with glossary bonuses reduces false omissions on domain terms (Aiba et al., 2024); WER and missing-keyword flags per slide are the primary metrics. The integration of these diagnostic methods into a complete coaching pipeline is detailed in Appendix A.

6 Datasets and Evaluation

The diagnostic methods described above require annotated data for training and evaluation. Table 2 summarizes major corpora and assesses their suitability for presentation coaching according to whether the following are present in the corpus: whether the learners are L2 speakers, long-form speech characteristic of presentations, accent or L1 annotations, and prosody labels (e.g. pitch accent).

As Table 2 shows, no existing corpus simultaneously provides L2 speech, long-form presentation structure, accent labels, and prosodic annotations. TIMIT and L2-ARCTIC focus on isolated sentences; TED-LIUM 3 and GigaSpeech (Chen et al., 2021) are long-form but not L2-specific; Common Voice provides accent labels but consists of short read sentences. Newer resources partially close the gap: EpaDB (Vidal et al., 2019) offers detailed phone-level pronunciation annotations for L2 Spanish speakers of English, and the recently released Speak & Improve Corpus (Knill et al., 2024) provides approximately 340 hours of spontaneous L2 English speech with CEFR proficiency scores and diverse L1 backgrounds. However, neither includes slide structure or presentation-specific annotations. This gap directly limits the ability to train and evaluate systems on the dimensions most critical for presentation coaching, particularly discourse prosody and slide-aligned pacing. We recommend creating small, controlled *presentation mini-sets*, consisting of 5–10 minute talks per speaker with slide markers, emphasis annotations, and diverse accent coverage, as a near-term community benchmark.

To evaluate coaching systems using such data, we consolidate metrics across five categories (Table 3), each aligned with one or more taxonomy dimensions. Importantly, the individual metrics we recommend are not novel; they are established measures drawn from prior work in prosody analysis, fluency assessment, and speech quality evaluation. Our contribution lies in their systematic alignment with the five taxonomy dimensions to form a coherent evaluation framework for presentation coaching. This alignment ensures that each taxonomy dimension has clearly defined, measurable targets, enabling researchers to evaluate coaching systems comprehensively rather than on isolated aspects.

Segmental metrics assess phone- and word-level pronunciation accuracy. Phone/word F1 measures the precision and recall of correctly produced segments, while unweighted average recall (UAR) ensures that performance on rare error types is not masked by frequent correct phones. Both are derived from GOP (Witt and Young, 2000) or CTC-based (Cao et al., 2024) diagnostic outputs.

Prosody metrics capture how well a learner’s intonation matches the reference. $\log-F_0$ RMSE quantifies average pitch deviation, and Pearson r measures the correlation of pitch contours, reflecting whether the learner’s intonation follows the same rises and falls as the target (Rosenberg, 2010;

Category	Key Metrics	Diagnostic Method
Segmental	Phone/Word F1, UAR	GOP (Witt and Young, 2000), CTC
Prosody	Log- F_0 RMSE, Pearson r	Pitch track + DTW
Pacing	WPM deviation, Pause rate	Alignment boundaries
Faithfulness	WER, Glossary hit-rate	Constrained ASR
Perceptual	MOS, PESQ	Human / Neural pred.

Table 3: Evaluation metrics for presentation coaching, organized by taxonomy dimension. We adopt individual metrics from prior work; we consolidate them into a unified framework aligned with the five coaching dimensions.

Zahorian and Hu, 2008).

Pacing metrics evaluate temporal control. WPM deviation measures how far the learner’s speaking rate falls from the target band, and pause rate captures whether pauses occur at appropriate boundaries (commas, slide breaks) rather than mid-phrase (Shen et al., 2021; Hincks, 2005).

Faithfulness metrics assess content coverage. WER measures overall transcription accuracy against the script (Radford et al., 2023), while glossary hit-rate tracks whether domain-critical terms are produced correctly, using constrained ASR with glossary bonuses (Aiba et al., 2024).

Perceptual metrics provide a holistic quality check. Mean Opinion Score (MOS) captures human judgments of overall speech quality, and PESQ (Rix et al., 2001) offers an automated proxy. Neural MOS predictors (Lo et al., 2019) can further scale these assessments without requiring human listeners for every evaluation.

Metrics alone are insufficient without evidence that they translate into actual learner improvement, and its underlying diagnostics have to remain reliable in realistic conditions. We therefore recommend three complementary validation protocols. First, **shadowing efficacy** should be tested via randomized crossover studies comparing TTS-guided (Chen et al., 2024) and unguided rehearsal, measuring whether exemplar-based practice yields measurable gains on the five taxonomy dimensions. Second, **diagnostic precision** should be evaluated by comparing GOP (Witt and Young, 2000) and clone-and-compare methods against expert annotations, since the choice of diagnostic directly affects the feedback a learner receives. Third, **robustness** should be assessed by measuring metric drift under environmental noise at multiple SNR levels, because real rehearsal settings rarely match studio-quality recording conditions.

7 Open Problems and Research Opportunities

The preceding sections reveal that while individual components of presentation coaching (pronunciation diagnostics, prosody modeling, exemplar generation) have matured considerably, integrating them into comprehensive, deployable systems remains an open challenge. Rather than restating these gaps, we identify five concrete research directions with specific actionable proposals.

Closing the lexical stress and discourse prosody gap. Lexical stress remains the most under-addressed taxonomy dimension (Table 1), yet mis-stressed words are a leading cause of reduced L2 comprehensibility (Munro and Derwing, 1995). A concrete next step is to extend self-supervised mispronunciation detectors (Xu et al., 2021; Gong et al., 2022) with syllable-level prominence classifiers trained on forced-aligned lexical stress annotations. For discourse prosody, current systems assess intonation at the utterance level, but presentations require section-level prosodic planning (e.g., pitch resets at topic boundaries, rising contours for rhetorical questions) (Hirschberg, 2004; Xu, 2005). We propose training prosodic planning models that condition on slide structure and discourse markers, enabling feedback such as “lower your pitch at the start of a new section.”

Real-time, personalized coaching. Only two of 15 surveyed systems provide real-time feedback (Table 1), yet immediate feedback is pedagogically more effective than post-hoc reports (Golonka et al., 2014). Sub-second diagnostics during live rehearsal require neural audio codecs (Défossez et al., 2022) and efficient alignment algorithms (e.g., sub-sequence DTW or streaming CTC). Beyond latency, systems should learn user-specific pacing bands and prosodic targets from minimal enrollment data rather than enforcing fixed norms (Trofimovich and Baker, 2006). A concrete proposal is to develop few-shot personalization modules that adapt pacing and prosody targets from a 2–3 minute calibration recording, enabling coaching that improves clarity without erasing the speaker’s natural style.

Evaluation infrastructure and benchmark creation. The field lacks standardized corpora with slide-aligned scripts, emphasis annotations, and diverse L1 coverage (Section 6). We recommend a community effort to create *presentation mini-sets*:

20–30 speakers from 5+ L1 backgrounds, each delivering a 5–10 minute technical talk with time-stamped slide boundaries, phone-level transcriptions, and expert prosody ratings. Such a benchmark would enable head-to-head comparison of coaching systems on all five taxonomy dimensions and support shared evaluation campaigns analogous to SUPERB (Yang et al., 2021) for speech processing. Equally important is validating that proposed metrics predict actual learner improvement, not just reference similarity; longitudinal studies measuring pre/post coaching gains on each taxonomy dimension remain rare and are critically needed.

LLM-augmented coaching and multimodal integration. The success of LLM-based coaching (Aiba et al., 2024; Chen et al., 2025) raises broader questions about how large language models can be integrated across all five taxonomy dimensions. Concrete opportunities include: (a) generating contextualized drill prompts (“Try saying *algorithm* with stress on the first syllable”) that target specific taxonomy dimensions identified by diagnostics; (b) evaluating content faithfulness through semantic similarity rather than word-level WER, enabling tolerance for paraphrasing while still catching substantive omissions; and (c) adapting coaching tone to individual learner profiles and proficiency levels. Recent work shows that speech LLMs can outperform prior baselines for L2 oral proficiency assessment (Ma et al., 2025), suggesting a path toward end-to-end systems that jointly assess pronunciation, prosody, and fluency. However, grounding LLM outputs in acoustically verified evidence, rather than generating plausible-sounding but inaccurate feedback, remains a key challenge. Future systems should also incorporate visual cues (slides, gestures) via vision-language models (Radford et al., 2021; Girdhar et al., 2023) to enable truly multimodal coaching. Extending coaching beyond English to other academic presentation languages (e.g., Mandarin-accented English, multilingual code-switching) requires corpora, phoneme sets, and G2P systems that do not yet exist at scale (Conneau et al., 2021).

Bridging the research and industry gap. Commercial L2 pronunciation and presentation coaching applications have grown rapidly. Platforms such as ELSA Speak and Speechling target segmental pronunciation, while Yoodli and Orai focus on pacing and filler-word reduction for general

public speaking. However, a significant gap exists between research prototypes and deployed systems. Commercial platforms typically address one or two taxonomy dimensions and rarely publish their technical approaches, making systematic comparison with research systems difficult. Furthermore, industry systems face practical constraints, including device heterogeneity, diverse noise environments, and the need for sub-second feedback on mobile hardware, that are rarely addressed in academic evaluations. We recommend that: (a) future surveys explicitly compare commercial and research systems where possible; (b) researchers engage with deployed platforms to identify real-world failure modes that controlled experiments may miss; and (c) standardized evaluation protocols be developed that span both research and commercial systems. The fragmentation between research (which advances methods) and industry (which addresses deployment constraints) could be reduced through shared benchmarks and open evaluation challenges.

8 Conclusion

We have surveyed automated presentation coaching systems, organizing them through a five-dimensional taxonomy (pronunciation, lexical stress, prosody, pacing, and content faithfulness) and mapping 15 representative systems onto this framework. Our comparison (Table 1) reveals that lexical stress and content faithfulness are dramatically under-addressed, that real-time feedback remains rare, and that no existing system integrates all five dimensions. We reviewed the core technical methods, including TTS-based exemplar generation and diagnostic approaches such as GOP/CTC, clone-and-compare, and prosody/pacing metrics, and showed how they relate to the taxonomy. Open challenges include the scarcity of annotated presentation corpora, accent-fair evaluation across diverse L1 backgrounds, the latency constraints of real-time coaching, and the gap between research prototypes and industry deployment. We hope this survey and its associated taxonomy serve as a useful reference for researchers developing the next generation of integrated, evidence-based presentation-coaching systems.

Limitations

This survey has several limitations that should be acknowledged. First, while we aim to cover representative work at the intersection of CAPT, TTS,

and L2 speaking systems, our coverage is not exhaustive; for space reasons we emphasize studies that directly inform pronunciation, prosody, pacing, and comprehensibility for slide-based L2 English presentations. Second, while we synthesize methods from TTS and CAPT domains, we focus primarily on English presentation training; the applicability to other languages and presentation styles (e.g., storytelling, persuasive speaking) requires further investigation. Third, our taxonomy and metrics are derived from existing literature and may not capture all dimensions relevant to real-world coaching scenarios. Fourth, the datasets we curate and recommend are limited in scale; large-scale presentation corpora with diverse accents and domains would strengthen evaluation. Fifth, we discuss voice cloning for personalized references but do not provide empirical comparisons of clone-and-compare vs. expert references across different speaker populations. Finally, while we address ethics and fairness, the practical implementation of accent-fair thresholds and privacy-preserving systems requires more extensive validation in deployed settings.

Ethics Statement

As automated coaching systems become more widely deployed, ethical concerns around fairness, privacy, and data governance are core design requirements. Many surveyed systems collect sensitive voice data and generate personalized feedback, making the following principles essential for trustworthy deployment.

Accent fairness. Calibrate thresholds per L1 cohort to account for systematic phonetic differences (Flege, 1995; Best and Tyler, 2007); report subgroup metrics (e.g., precision/recall by L1) to identify potential biases; and avoid penalizing identity-marking phonetic features (e.g., rhoticity patterns, vowel quality (Wells, 1982; Ladefoged and Johnson, 2015)) when intelligibility is unaffected (Munro and Derwing, 1995; Derwing and Munro, 2005). Systems should help learners improve clarity without erasing accent identity (Isaacs and Trofimovich, 2018).

Voice cloning and data governance. Obtain explicit consent before creating voice clones, clearly explaining how embeddings will be used and stored; encrypt voice embeddings both in transit and at rest; support deletion/expiry mechanisms;

and allow non-clone baselines (e.g., generic TTS or expert references) for users who opt out. For data governance, disclose TTS licenses, avoid retaining raw audio longer than needed (consider on-device processing), and clearly communicate data collection practices.

Prosodic perception. Listener-impression studies (Shoda et al., 2023) highlight that minor pitch/timing adjustments can change perceived competence and speaker traits. Ethical coaching systems should avoid reinforcing biased mappings between prosody, accent, and personas, presenting feedback as optional stylistic guidance rather than prescriptive judgments.

Acknowledgments

We thank Ziwei Gong for reviewing the paper and providing valuable feedback.

References

- Mayuko Aiba, Daisuke Saito, and Nobuaki Minematsu. 2024. [A chatgpt-based oral q&a practice system for first-time student participants in international conferences](#). In *Proceedings of the 25th Annual Conference of the International Speech Communication Association (Interspeech 2024)*, Kos, Greece. ISCA.
- Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proc. LREC*, pages 4218–4222.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems* 33.
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.
- Catherine T. Best and Michael D. Tyler. 2007. Nonnative and second-language speech perception: Commonalities and complementarities. In *Language Experience in Second Language Speech Learning: In Honor of James Emil Flege*, pages 13–34. John Benjamins.
- Xinwei Cao, Zijian Fan, Torbjørn Svendsen, and Giampiero Salvi. 2024. A framework for phoneme-level pronunciation assessment using etc. In *Proc. Interspeech*.

- Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. 2022. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *Proc. ICML*, pages 2709–2720.
- Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. 2021. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *InterSpeech 2021*, pages 3670–3674.
- Lei Chen, Gary Feng, Chee Wee Leong, Blair Lehman, Michelle Martin, Hannah Graf, Aaron Clauset, and Su-Youn Yoon. 2014. Automated evaluation of presentation skills using speech recognition and audience feedback. In *Proc. Workshop on Intelligent Tutoring Systems for Ill-Defined Domains*, pages 1–10.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.
- Sirui Chen, Jinsong Zhou, Xinli Xu, Xiaoyu Yang, Litao Guo, and Ying-Cong Chen. 2025. Present-coach: Dual-agent presentation coaching through exemplars and interactive feedback. *arXiv preprint arXiv:2511.15253*.
- Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024. *F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching*. *arXiv preprint arXiv:2410.06885*.
- Cameron Churchwell, Max Morrison, and Bryan Pardo. 2024. High-fidelity neural phonetic posteriorgrams. *arXiv preprint arXiv:2402.17735*.
- Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Un-supervised cross-lingual representation learning for speech recognition. In *Interspeech 2021*, pages 2426–2430.
- Catia Cucchiari, Ambra Neri, and Helmer Strik. 2009. Oral proficiency training in dutch L2: The contribution of ASR-based corrective feedback. *Speech Communication*, 51(10):853–863.
- Ionut Damian, Chiew Seng Sean Tan, Tobias Baur, Johannes Schöning, Kris Luyten, and Elisabeth André. 2015. Logue: A real-time feedback system for nonverbal presentation skills. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1835–1840.
- Steven Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.
- Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.
- Tracey M. Derwing and Murray J. Munro. 2005. Second language accent and pronunciation teaching: A research-based approach. *TESOL Quarterly*, 39(3):379–397.
- Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jingren Zhou. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.
- James Emil Flege. 1995. Second language speech learning: Theory, findings, and problems. In Winifred Strange, editor, *Speech Perception and Linguistic Experience: Issues in Cross-Language Research*, pages 233–277. York Press.
- Horacio Franco, Leonardo Neumeyer, Yoon Kim, and Orith Ronen. 1997. Automatic pronunciation scoring for language instruction. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1471–1474.
- Tania Amanda Nkoyo Frederick Eneye, Chukwuebuka Fortunate Ijezue, Ahmad Imam Amjad, Maaz Amjad, Sabur Butt, and Gerardo Castañeda-Garza. 2025. *Advances in auto-grading with large language models: A cross-disciplinary survey*. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 477–498. Association for Computational Linguistics.
- John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren. 1993. TIMIT acoustic-phonetic continuous speech corpus. LDC93S1.
- Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Manat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proc. CVPR*, pages 15180–15190.
- Ewa M. Golonka, Anita R. Bowles, Victor M. Frank, Dorna L. Richardson, and Suzanne Freynik. 2014. Technologies for foreign language learning: A review of technology types and their effectiveness. *Computer Assisted Language Learning*, 27(1):70–105.

- Yuan Gong, Ziyi Chen, Iek-Heng Chu, Peng Chang, and James Glass. 2022. Transformer-based multi-aspect multi-granularity non-native english speaker pronunciation assessment. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, pages 369–376.
- Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pages 5036–5040.
- Yo Hamada. 2018. Shadowing for pronunciation development: Haptic-shadowing and IPA-shadowing. *Journal of Asia TEFL*, 15(1):167–183.
- François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. 2018. Tedlium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Proc. SPECOM*, pages 198–208.
- Rebecca Hincks. 2005. *Computer Support for Learners of Spoken English*. Ph.D. thesis, KTH Royal Institute of Technology.
- Julia Hirschberg. 2004. Pragmatics and intonation. In Laurence R. Horn and Gregory Ward, editors, *The Handbook of Pragmatics*, pages 515–537. Blackwell.
- T. Hori. 2008. *Exploring Shadowing as a Method of English Pronunciation Training*. Ph.D. thesis, Kwansei Gakuin University, Nishinomiya, Japan.
- Chih-Chieh Hsieh, Damin Dong, and Hsien-Chin Wang. 2013. A preliminary study of applying shadowing technique to english intonation instruction. *Taiwan Journal of Linguistics*, 11(2):43–66.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.
- Talia Isaacs and Pavel Trofimovich. 2018. Cognition, intelligibility, and the boundaries of phonology: A look at l2 speech. *Annual Review of Applied Linguistics*, 38:125–143.
- Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Proc. NeurIPS*, pages 4485–4495.
- Daniel Jurafsky and James H. Martin. 2026. *Speech and language processing*. Online manuscript, Jan. 6, 2026 release.
- Shuhei Kadota. 2019. *Shadowing as a Practice in Second Language Acquisition: Connecting Inputs and Outputs*. Routledge.
- Kate Knill, Diane Nicholls, Mark J.F. Gales, Mengjie Qian, and Pawel Strojnski. 2024. Speak & improve corpus 2025: An L2 english speech corpus for language assessment and feedback. *arXiv preprint arXiv:2412.11986*.
- Daniel Korzekwa, Roberto Barra-Chicote, Szymon Zaporowski, Grzegorz Beringer, Jaime Lorenzo-Trueba, Jasha Droppo, Thomas Drugman, and Bożena Kostek. 2021. Detection of lexical stress errors in non-native (L2) english with data augmentation and attention. *arXiv preprint arXiv:2012.14788*.
- Peter Ladefoged and Keith Johnson. 2015. *A Course in Phonetics*, 7th edition. Cengage Learning.
- Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. In *Advances in Neural Information Processing Systems* 36.
- Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. 2019. MOSNet: Deep learning-based objective assessment for voice conversion. In *Proc. Interspeech*, pages 1541–1545.
- Stephen E. Lucas. 2014. *The Art of Public Speaking*, 12th edition. McGraw-Hill.
- Rao Ma, Mengjie Qian, Siyuan Tang, Stefano Banno, Kate M. Knill, and Mark J.F. Gales. 2025. Assessment of L2 oral proficiency using speech large language models. In *Proc. Interspeech*.
- Sherwyn Morreale, Philip Backlund, and Leyla Sparks. 2014. Communication education and instructional communication: Genesis and evolution as fields of inquiry. *Communication Education*, 63(4):344–354.
- Murray J. Munro and Tracey M. Derwing. 1995. Foreign accent, comprehensibility, and intelligibility in the speech of second language learners. *Language Learning*, 45(1):73–97.
- Ambra Neri, Catia Cucchiari, Helmer Strik, and Lou Boves. 2002. The pedagogy-technology interface in computer assisted pronunciation training. *Computer Assisted Language Learning*, 15(5):441–467.
- Kentaro Onda, Keisuke Imoto, Satoru Fukayama, Daisuke Saito, and Nobuaki Minematsu. 2025. Prosodically enhanced foreign accent simulation by discrete token-based resynthesis only with native speech corpora. In *Proc. Interspeech*.

- Kentaro Onda, Joonyong Park, Nobuaki Minematsu, and Daisuke Saito. 2024. [A pilot study of GSLM-based simulation of foreign accentuation only using native speech corpora](#). In *Interspeech 2024*, pages 3600–3604.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proc. ICASSP*, pages 5206–5210.
- Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. 2019. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *Proc. ICML*, pages 5210–5219.
- Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*, pages 28492–28518.
- Vikram Ramanarayanan, Lei Chen, Chee Wee Leong, Gary Feng, and David Suendermann-Oeft. 2015. An analysis of time-aggregated and time-series features for scoring different aspects of multimodal presentation data. In *Proc. Interspeech*.
- Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. FastSpeech 2: Fast and high-quality end-to-end text to speech. In *Proc. ICLR*.
- Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. 2001. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *Proc. ICASSP*, volume 2, pages 749–752.
- Andrew Rosenberg. 2010. Autobi: A tool for automatic tobi annotation. In *Proc. Interspeech*, pages 146–149.
- Kazuya Saito, Konstantinos Macmillan, Magdalena Kachlicka, Takuya Kuniyara, and Nobuaki Minematsu. 2023. Automated assessment of second language comprehensibility: Review, training, validation, and generalization studies. *Studies in Second Language Acquisition*, 45(1):234–263.
- Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.
- Jan Schneider, Dirk Börner, Peter van Rosmalen, and Marcus Specht. 2015. Presentation trainer, your public speaking multimodal coach. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 539–546.
- Yang Shen, Ayano Yasukagawa, Daisuke Saito, Nobuaki Minematsu, and Kazuya Saito. 2021. [Optimized prediction of fluency of L2 english based on interpretable network using quantity of phonation and quality of pronunciation](#). In *2021 IEEE Spoken Language Technology Workshop (SLT)*.
- Chihiro Shoda, Yingxiang Gao, Yurun He, Nobuaki Minematsu, Noriko Nakanishi, and Daisuke Saito. 2023. [Learners’ prosodic control in the task of expressive storytelling and predicted native listeners’ impressions of the learners’ speech](#). In *9th Workshop on Speech and Language Technology in Education (SLaTE 2023)*, pages 46–50.
- Helmer Strik, Khiet Truong, Febe de Wet, and Catia Cucchiari. 2009. Comparing different approaches for automatic pronunciation error detection. *Speech Communication*, 51(10):845–852.
- M. Iftexhar Tanveer, Emy Lin, and Mohammed Ehsan Hoque. 2015. Rhema: A real-time in-situ intelligent interface to help people with public speaking. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 286–295.
- Pavel Trofimovich and Wendy Baker. 2006. Learning second language suprasegmentals: Effect of 12 experience on prosody and fluency characteristics of 12 speech. *Studies in Second Language Acquisition*, 28(1):1–30.
- Jazmín Vidal, Luciana Ferrer, and Leonardo Brambilla. 2019. EpaDB: A database for development of pronunciation assessment systems. In *Proc. Interspeech*, pages 589–593.
- Steven H. Weinberger. 2015. Speech accent archive. *George Mason University*. <http://accent.gmu.edu>.
- John C. Wells. 1982. *Accents of English*. Cambridge University Press.
- Silke M. Witt. 2012. Automatic error detection in pronunciation training: Where we are and where we need to go. *Proc. International Symposium on Automatic Detection of Errors in Pronunciation Training (ISADEPT)*, pages 1–8.
- Silke M. Witt and Steve J. Young. 2000. Phonetic segment evaluation for automatic assessment of pronunciation quality. In *Proc. ICSLP*.
- Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, and Long Ma. 2021. Explore wav2vec 2.0 for mispronunciation detection. In *Interspeech 2021*, pages 4428–4432.

Yi Xu. 2005. Speech melody as articulatorily implemented communicative functions. *Speech Communication*, 46(3-4):220–251.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhota, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Kotik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. 2021. SUPERB: Speech processing universal performance benchmark. In *Interspeech 2021*, pages 1194–1198.

Stephen A. Zahorian and Hongbing Hu. 2008. A spectral/temporal method for robust fundamental frequency tracking. *The Journal of the Acoustical Society of America*, 123(6):4559–4571.

Klaus Zechner, Derrick Higgins, Xiaoming Xi, and David M. Williamson. 2009. Automatic scoring of non-native spontaneous speech in tests of spoken english. *Speech Communication*, 51(10):883–895.

Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yukai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. speechocean762: An open-source non-native english speech corpus for pronunciation assessment. In *Interspeech 2021*, pages 3710–3714.

Guanlong Zhao, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. 2018. L2-ARCTIC: A non-native english speech corpus. In *Interspeech 2018*, pages 2783–2787.

A System Architecture and UI Patterns

Translating the methods and metrics discussed above into a usable coaching system requires careful attention to architecture, latency, and user interface design (Hincks, 2005; Chen et al., 2014). This section outlines practical technical considerations drawing on principles from public speaking pedagogy (Lucas, 2014; Morreale et al., 2014) and computer-assisted language learning (Golonka et al., 2014).

A typical coaching pipeline consists of seven stages: (1) script ingest & glossary extraction; (2) exemplar synthesis (anchor/target bands, emphasis) using modern TTS systems (Chen et al., 2024); (3) user recording in short chunks; (4) alignment (CTC timing (Graves et al., 2006); DTW (Sakoe and Chiba, 1978) fallback); (5) diagnostics (segmental GOP (Witt and Young, 2000)/DTW + prosody/pacing); (6) drill generation; and (7) progress tracking with per-slide thresholds. This architecture generalizes patterns found in prior L2 speaking and presentation systems (Aiba et al., 2024; Shen et al.,

2021; Saito et al., 2023), clarifying the interaction between TTS, diagnostics, and UI components.

To support interactive rehearsal, systems must meet strict latency constraints, targeting sub-second feedback per chunk. This includes TTS rendering in under 200ms for 5–8s text (achievable with non-autoregressive models like F5-TTS (Chen et al., 2024)), alignment and metric computation within 300ms, and UI updates in under 200ms. Privacy-sensitive applications should prioritize on-device inference and defer computationally intensive voice cloning to an offline enrollment step with explicit consent.

Finally, effective coaching interfaces should visualize these metrics through multiple synchronized views: (1) **waveform overlay** with log- F_0 contours (extracted using robust pitch trackers (Zahorian and Hu, 2008)) for comparing learner and reference prosody; (2) **word-level heatmap** color-coded by pronunciation scores (GOP (Witt and Young, 2000) or DTW (Sakoe and Chiba, 1978) distance) to identify problematic segments; (3) **twin playheads** for scrubbing corresponding moments in user and reference audio; (4) **per-slide checklist** highlighting top issues (e.g., “stress on *algorithm*”); (5) **pacing gauge** showing band compliance; and (6) **one-click drills** that auto-loop on error spans for focused practice.

The Aftermath of DrawEduMath: Vision Language Models Underperform with Struggling Students and Misdiagnose Errors

Li Lucy[△] Albert Zhang⁺ Nathan Anderson[÷] Ryan Knight⁺ Kyle Lo^{□△}

[△]University of Washington ⁺Insource Services [÷]Worcester Polytechnic Institute

[□]Allen Institute for AI

lucy3li@cs.washington.edu kylel@allenai.org

Abstract

Effective mathematics education requires identifying and responding to students' mistakes. For AI to support pedagogical applications, models must perform well across different levels of student proficiency. Our work provides an extensive, year-long snapshot of how 11 vision-language models (VLMs) perform on DrawEduMath, a QA benchmark involving real students' handwritten, hand-drawn responses to math problems. We find that models' weaknesses concentrate on a core component of math education: student error. All evaluated VLMs underperform when describing work from students who may require more pedagogical help, and across all QA, they struggle the most on questions related to assessing student error. Thus, while VLMs may be optimized to be math problem solving experts, our results suggest that they require alternative development incentives to adequately support educational use cases.

1 Introduction

The use of vision language models (VLMs) in education has received increasing attention in both academic research (Küchemann et al., 2025; Lee et al., 2025b) and commercial AI products. Examples of the latter include Google Classroom with Gemini integration,¹ and Khan Academy's AI tutor Khanmigo,² powered by OpenAI models. However, the integration of these models into tutoring and classroom settings often lacks transparent, open, and realistic evaluation. With this gap in mind, we previously released DrawEduMath (Figure 1), a dataset consisting of 2,030 teacher-annotated images of real students' hand-drawn responses to K-12 math problems (Baral et al., 2025). In contrast to other multimodal math understanding

or problem solving benchmarks (Alshammari et al., 2026), DrawEduMath involves noisy, naturalistic data pulled from an online learning platform (Figure 1). In the year since its release, we continuously updated the benchmark's leaderboard³ with newer models.

Our paper offers a snapshot of how 11 VLMs have performed on DrawEduMath in the year after its release (Figure 2).⁴ We surface two key findings:

F1: VLMs are worse at describing the contents of student work that contains math errors than student work without errors.

F2: VLMs still struggle the most on question types related to assessing students' correctness.

These findings suggest that VLMs underperform with students who need additional pedagogical support (**F1**), and they also fail to appropriately identify cases when support is needed (**F2**).

To investigate these patterns further, we conduct five analyses in §4-§8 targeting factors that may relate to model performance. Our experiments show that the performance gap in **F1** persists even when controlling for problem (§4) and when image noise is reduced (§5). In addition, a possible explanation for **F1** is that VLMs expect mathematically correct input images. Indeed, we find that some models' wrongly predicted answers for erroneous student work are similar to gold answers for non-erroneous student work (§6).

We also find that models do improve in assessing student correctness (**F2**) when provided gold natural language descriptions of student work (§7). However, their performance on these questions with extra textual support still lags behind their out-of-the-box performance on other question types. Finally, though models seemingly perform better

¹<https://blog.google/outreach-initiatives/education/classroom-ai-features/>

²<https://www.khanmigo.ai/>

³<https://drawedumath.org/>

⁴DrawEduMath was released to accompany an initial non-archival paper in Dec 2024 (Baral et al., 2024, 2025).

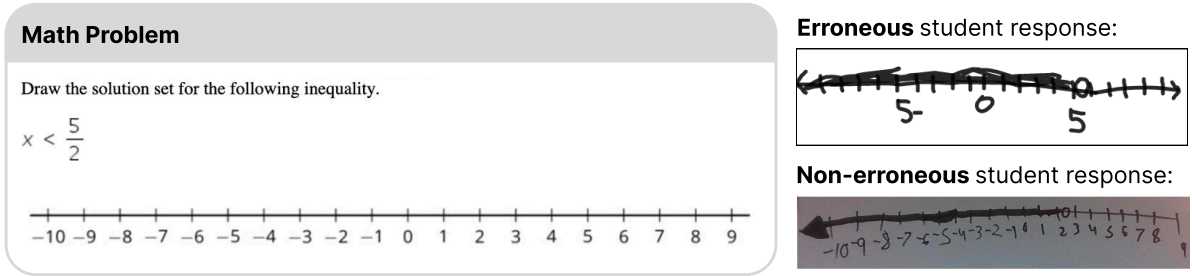


Figure 1: On the left is a math problem, where students are asked to draw $x < 5/2$ on a number line. The right side shows two example student responses that differ in correctness. DrawEduMath pairs each math problem with one student response, and prompts VLMs to answer questions about the student response.

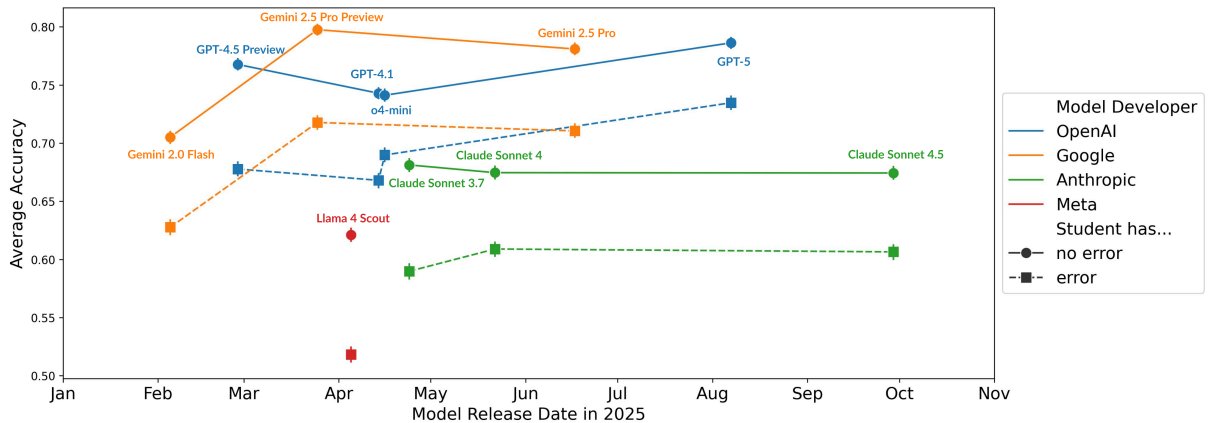


Figure 2: VLMs consistently perform worse on answering DrawEduMath benchmark questions pertaining to erroneous student responses. Performance on non-erroneous student responses (○) is labeled with specific VLMs’ names; that same model’s performance on erroneous student responses is directly below (□). Error bars are 95% CI.

on binary correctness questions (e.g. “Does the student do ___ correctly?”) than open-ended ones (e.g. “What errors does the student make in their response?”), some VLMs’ performance can sometimes be barely better than chance (§8).

Altogether, our in-depth error analysis of VLMs’ performance on real student math responses provides a clearer picture of their current weaknesses in supporting K-12 math education. We release data and scripts for reproducing our findings.⁵

2 Background & Related Work

Multimodal math benchmarks. Mathematical content, rich with diagrams, is ripe for evaluating models’ multimodal abilities. Thus, many vision-language benchmark creation efforts have targeted math (Alshammari et al., 2026; Zhang et al., 2024; Lu et al., 2024; Yan et al., 2025). Educational settings offer additional challenges, where VLMs may be assessed on their abilities to make higher-

level pedagogical inferences and handle handwritten, hand-drawn content (Parsaeifard et al., 2025; Latif et al., 2025; Nath et al., 2025; Nguyen et al., 2025). For example, MathCog asks models to diagnose students’ cognitive skills using binary yes/no questions and a digitally handwritten dataset (Jin et al., 2025). Within this landscape of prior work, DrawEduMath remains a significant evaluation resource, given its diversity of image and question types, its use of noisy, real student work, and its inclusion of experienced teacher annotations (Baral et al., 2025).

AI & student error. Student mistakes and misconceptions have long been a focal point in education research (Smith III et al., 1994; Radatz, 1979; Borasi, 1994; Metcalfe, 2017). With increased attention towards AI as tutors and teaching assistants, research has focused on models’ abilities to identify student error (Srivatsa et al., 2025; Kochmar et al., 2025; Daheim et al., 2024), reason about patterned misconceptions (Rittle-Johnson et al., 2025; Ross and Andreas, 2025), correct errors (Mita et al.,

⁵https://github.com/lucy3/aftermath_drowedumath

2024), and provide feedback (Kaliisa et al., 2026; Botelho et al., 2023; Stahl et al., 2024). There is some research around model robustness to user error, but much of it focuses on linguistic errors in prompts (Gan et al., 2024; Chatterjee et al., 2024). There is little work on how math errors impact model performance: one example is Daheim et al. (2024), who show that language models are worse at verifying the correctness of erroneous student math than non-erroneous math. Our work reaches a similar conclusion across more QA types and with multimodal data, though our results around models’ correctness & error assessments paint a less straightforward picture (§8).

Risks of AI in education. The field of AI & education could be considered a form of AI for “social good” (Cowls et al., 2021), driven by goals that counter AI’s negative impacts on human skill formation and cognitive thinking (Bastani et al., 2025; Klimova and Pikhart, 2025; Shen and Tamkin, 2026; Lee et al., 2025a). However, though its end-goals may be optimistically framed, AI in education is not without risk (Blodgett and Madaio, 2021; Holstein and Doroudi, 2021). Our work is aligned with literature that investigates how AI may disparately impact different student populations (Schaller et al., 2024; Capraro et al., 2024; Hadar Shoval, 2025); our distinct approach is that we group student inputs based on demonstrated math proficiency.

3 Evaluating VLMs with DrawEduMath

3.1 Data

DrawEduMath is an English-language dataset of 2,030 images of students’ handwritten responses to K-12 math problems (Baral et al., 2025). Images are provided by the online learning platform ASSISTments and contain math problems drawn from open educational resources (Heffernan and Heffernan, 2014; Feng et al., 2025). Each image includes a math problem on the left and a student’s response on the right, and is accompanied by three types of data:

1. **Free-form captions** (2.0k+) from teachers describing each student response image.
2. **Synthetic QA pairs** (44.4k+), produced by Claude-3.5 Sonnet and GPT-4o reformatting facets of teachers’ captions into QA, e.g. *On the left-hand side of the image, the student wrote the word syrup → What word did the*

student write on the left-hand side of the image? Syrup.

3. **Teacher-written QA pairs** (11.6k+). Teachers wrote a set of shared questions for each math problem, followed by answers for each student response to each problem. Teachers answered two additional generic questions, *What errors does the student make in their response?* and *What strategy does the student use to solve the problem?* across all problems and student responses.

DrawEduMath includes a taxonomy of seven question types. In our analysis, we simplify this taxonomy into three types: *image creation and medium* (12.3%), *correctness & errors* (8.5%), and *content description* (79.2%). The first two match the benchmark’s original taxonomy, while the third question type is an aggregation of all other question types, ranging from the student’s problem solving strategy to the meaning, positioning, and frequency of drawn/written elements. We aggregate these question types in our main text, because our main findings generalize across more fine-grained types (Appendix A). The original DrawEduMath paper includes additional dataset statistics and QA examples (Baral et al., 2025).

3.2 Evaluation Setup

Scoring metric. Baral et al. (2025) used Mixtral 8x22B to judge the similarity of VLMs’ generated answers to gold answers on a scale of 1 (*quite different answers*) - 4 (*basically the same*), and then binarized these ratings when computing models’ accuracy. As LMs update over time, older ones like Mixtral 8x22B become deprecated in model API services. Thus, all evaluation in our work uses an updated LLM judge. We take the majority vote from three judges: Claude Sonnet 4.5, Gemini 2.5 Pro, and GPT-4o.⁶ Our updated judge achieves similar correlation (Spearman $\rho = 0.808$) with the same set of human judgements as Baral et al. (2025)’s original judge ($\rho = 0.801$). We compute and report binarized model accuracies; 1-2 are counted as incorrect, and 3-4 are correct.

Models. We evaluate 11 VLMs released in 2025 on DrawEduMath. Models span four developers: Open AI (GPT-4.1, GPT-4.5 Preview, o4-mini,

⁶Though GPT-4o is an older model, it has higher individual correlation with human annotations than the two newer models. So, we included it in our set of judges.

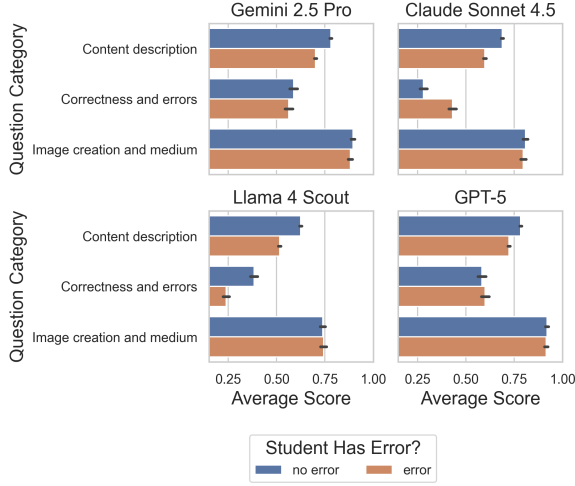


Figure 3: Content description QA consistently drives the gap in VLM performance between student responses that contain errors versus those that do not. Appendix A includes additional VLMs that expand this finding.

GPT-5), Anthropic (Claude Sonnet 3.7, Claude Sonnet 4, Claude Sonnet 4.5), Google (Gemini 2.0 Flash, Gemini 2.5 Pro, Gemini 2.5 Pro Preview), and Meta AI (Llama 4 Scout). Though our main findings **F1** & **F2** pertain to all of these models, the analyses in §5-§8 focus on four representative models: Gemini 2.5 Pro, Claude Sonnet 4.5, GPT-5, and Llama 4 Scout.

Labeling student error. To categorize whether students’ math responses contain an error or not, we use teachers’ answers to the question, *What errors does the student make in their response?* We ask GPT-5-mini to interpret each open-ended answer and classify it as *yes*, as in, the teacher describes some error, or *no*, for when teachers’ answers are variations of *There is no error* or *The student did not make an error* (Appendix B.1). We validate this LM annotator on a manually checked random sample of 200 examples (F1 = 0.984, Cohen’s κ = 0.970).

3.3 Main Findings

On average, models tend to perform worse when the student response contains an error (Figure 2). We find that this pattern is mostly driven by content description QA (**F1**, Figure 3). In addition, a weakness reported by Baral et al. (2025) in older VLMs persists in newer ones: questions related to students’ correctness and errors are still the most difficult (**F2**). In the next few sections, we dive into five factors that we hypothesize to relate to these findings: problem effects (§4), image noise (§5),

Model	β_1
Gemini 2.0 Flash	0.0944***
Gemini 2.5 Pro Preview	0.0888***
Gemini 2.5 Pro	0.0889***
GPT-4.1	0.0865***
GPT-4.5 preview	0.0874***
o4-mini	0.0612***
GPT-5	0.0585***
Claude Sonnet 3.7	0.0922***
Claude Sonnet 4	0.0816***
Claude Sonnet 4.5	0.0842***
Llama 4 Scout	0.1013***

Table 1: Estimated effects of student correctness (β_1) on VLMs’ accuracy on content description QA, where all $p < 1.0^{-12}$.

problem response defaults (§6), visual understanding bottlenecks (§7), and question open-endedness (§8).

4 Models underperform on erroneous student responses even when controlling for problem

One possibility is that the model performance gap observed by **F1** is actually not affected by the presence or absence of student error, but rather by some math problems being more difficult for VLMs to understand. DrawEduMath contains 188 unique math problems targeting concepts ranging from geometry to fractions. These problems span multiple grade levels, and on average, each problem has 12.64 student responses (Baral et al., 2025). Here, we show that the effect of student error on VLMs’ content description QA performance is statistically significant even when controlling for problem.

We estimate an ordinary least squares regression with problem fixed effects:

$$y_{ij} = \beta_0 + \beta_1(s_{ij}) + u_j + \epsilon_{ij}$$

In the equation above, y_{ij} is the average score a model has across content description QA for a student response i and problem j , u_j is a fixed effect for each problem, and ϵ_{ij} is the residual. If the student response is correct, $s_{ij} = 1$, otherwise $s_{ij} = 0$. Table 1 presents β_1 values across VLMs. These values show that even after controlling for problem, non-erroneous student responses significantly correspond with higher VLM performance.

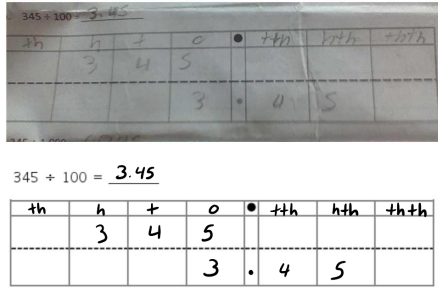


Figure 4: An example of how a student response image (top) is transformed and cleaned up by our digital redrawing process (bottom). This student uses a place value chart to show how digit values change for 345 after division by 100.

5 Models’ performance gaps are not strongly impacted by image noise

Another possible explanation for **F1** is that students who make math errors may simply submit noisier images. Students on ASSISTments may submit their answers by drawing digitally, or by uploading photographs of pen & paper work, which may include smudges and blur. In this section, we ask, does the model performance gap described by **F1** remain even when students’ responses are redrawn on a digital canvas in a standardized manner?

5.1 Experimental Setup

Redrawing images is a time-intensive process, requiring careful interpretation of each math problem and the intent of the original student response. So, for this experiment, we stratify sample one erroneous student response and one correct response from each problem, yielding 336 images in total. Though this sample is small, it retains statistically significant gaps in VLM performance on content description QA between erroneous and non-erroneous student response images (Table 2).

To encourage consistency, the lead author redrew all sampled student responses using a digital pen and the drawing application Procreate. This redrawing author retained students’ original positioning of content, and consulted teachers’ captions of images to navigate ambiguity and avoid faulty interpretation of problems and student responses. If needed, the author recreated elements such as graph paper grids and typed content in Figma. Figure 4 provides an illustrative example of how redrawing transforms students’ responses.

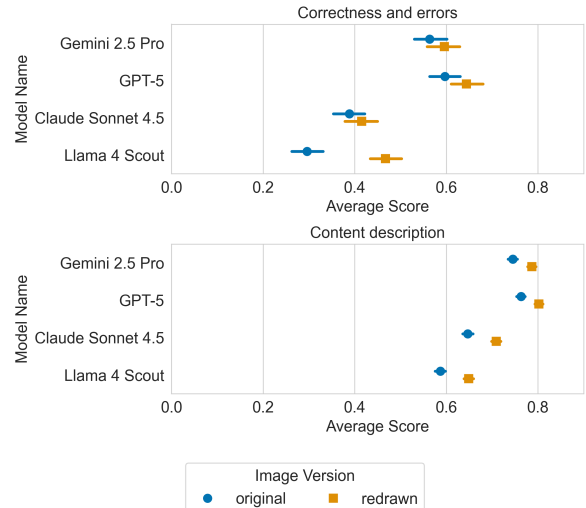


Figure 5: Models’ performance for content description QA generally improves after images are redrawn. Error bars are 95% CI.

Model	Original Images	Redrawn Images
Gemini 2.5 Pro	-0.089***	-0.096***
Claude Sonnet 4.5	-0.100***	-0.058**
GPT-5	-0.055**	-0.043**
Llama 4 Scout	-0.092***	-0.073***

Table 2: Differences in average scores on content description QA between erroneous and non-erroneous student images persist after redrawing. ** $p < 0.01$, *** $p < 0.001$.

5.2 Results

On redrawn images, VLMs’ performance shift in the expected direction, where scores generally improve with less image noise (Figure 5). Though requesting students to only submit born-digital content may make their work more interpretable for AI, not all classrooms have resources and policies that make such standardization feasible. In addition, pen-and-paper work remains vital, with studies showing that this traditional mode of learning can sometimes allow students to surpass their digital-only peers (Mueller and Oppenheimer, 2014; Altamura et al., 2025; Anthony et al., 2007; Umejima et al., 2021). Thus, one implication of Figure 5 is that the integration of VLMs in education may disparately impact analog and digital learners.

Importantly, we also find that models’ performance gap between erroneous and non-erroneous student images remains in redrawn images (Table 2). This result complements that of §4, by further isolating student error as a weak point in the use of VLMs in educational settings. Thus,

mitigation efforts around **F1** should focus on improving models’ understanding of erroneous mathematical content, across all levels of image noise and medium types.

6 Models default to assuming error-free math solutions

Why might erroneous student images be so challenging for VLMs? During a manual examination of VLMs’ QA errors, we observed that models sometimes produce plausible, though wrong, answers to benchmark questions, especially considering the context of the provided math problem (Figure 6). This observation suggests a possible explanation for **F1**: VLMs perform better on content description QA for error-free student responses, because models default to assuming error-free math solutions.

6.1 Analysis Setup

To quantify our observation using existing DrawEduMath annotations, we filter for content description QA shared across different student response images for the same math problem. Then, for each incorrect model answer to questions pertaining to erroneous student responses, we compare the model’s answer against true answers for non-erroneous student responses, and see whether the model’s incorrect answer matches the majority of these true ones. We compare model answers using the ensemble LM judge from §3.2. We only consider cases where we have at least two non-erroneous student response images associated with the given question, to ensure that we have sufficient signal of correct student behavior.

6.2 Results

Across four representative VLMs, incorrect model responses for erroneous student responses sometimes do match non-erroneous student solutions, with percentages ranging from 29% of content description QA mistakes for Gemini 2.5 Pro to 35% for Claude Sonnet 4.5 (Table 3). So, a sizable portion of benchmark answers may be inferable based on a math problem and a typical correct solution. There are many more ways a student response can be wrong than it can be correct, and so benchmark QA corresponding to correct student solutions navigate a narrower space of plausible possibilities.

Qualitatively, we observe that models especially tend to predict incorrect answers that match correct problem solutions when benchmark questions

Question How many dots did the student include in their array?	
For an erroneous student response:	
Model answer 12	True answer The student didn't include an array.
True answer for a non-erroneous student response: The student included 12 dots in their array.	

Question How many squares did the student draw to show the number of cups of red paint?	
For an erroneous student response:	
Model answer The student drew 9 squares to show the number of cups of red paint.	True answer The student drew 12 squares to represent the cups of red paint.
True answer for a non-erroneous student response: The student drew 9 squares to show the number of cups of red paint.	

Figure 6: Illustrative examples of the phenomenon where models predict answers for erroneous student responses that match true answers for non-erroneous students.

Model	%	<i>N</i>
Gemini 2.5 Pro	0.2923	1,355
Claude Sonnet 4.5	0.3519	1,921
GPT-5	0.3125	1,357
Llama 4 Scout	0.3060	2,134

Table 3: The percentage of times for which an incorrect model answer for a content description question and erroneous student image matched the majority of true answers for non-erroneous student images. *N* is the number of incorrect model answers in the denominator of the percentage.

involve false presuppositions. Figure 6 illustrates an example; there, the wording of the top teacher-written question assumes that the student has drawn any array at all. Teacher-written questions in DrawEduMath are those that teachers would like VLMs to answer across all student responses to a problem, mimicking potential uses of VLMs for learning analytics. Models’ susceptibility to false premises or suppositions is well-documented in prior work (e.g. Yu et al., 2023; Srikanth et al., 2024), and our work illustrates a consequence of this weakness for education-related applications.

Generally, language models are developed to be good math problem solvers. Math solving benchmarks are continuously emphasized in leaderboards and commercial LM releases (Cobbe et al., 2021; Hendrycks et al., 2021; Google DeepMind, 2025). To encourage mathematically correct outputs and hill-climb on these benchmarks, models are mostly exposed to “high quality”, correct math content during training (Mahabadi et al., 2025; Paster et al.,

2024). The challenge of understanding, but not generating faulty content has received attention in other domains. For example, toxicity is another case of an understanding vs. generation tradeoff; we want models that can detect, address, and understand toxic content, without generating it (Longpre et al., 2024; Wang et al., 2025). Our findings suggest that education is another domain where the application of alternative training methods on erroneous data, such as Wang et al. (2025), could be applicable.

7 Textual support can improve models’ correctness assessments to some extent

DrawEduMath QA range from low-level content description (e.g. “How many triangles did the student draw?”) to higher-level correctness judgments. Now, we move on from examining **F1**, which focuses on content description QA, to digging deeper into **F2**, which pertains to correctness & errors QA. Earlier, we saw that the latter remain difficult even after images are digitally cleaned up (Figure 5). Perhaps, image understanding is a bottleneck for models answering these more reasoning-intensive questions. To what extent can models improve their assessment of student error when given textual descriptions of student work?

7.1 Experimental Setup

As mentioned in §3.1, DrawEduMath includes teacher-written, gold captions of students’ response images. Baral et al. (2025) used these captions to synthetically generate a subset of the QA pairs in the benchmark. If a caption produces synthetic QA that fall in the correctness & error question category, we exclude that image from the current analysis to avoid input-output contamination.⁷ We append to each DrawEduMath input prompt these gold captions (prompt in Appendix C.1), and re-evaluate models’ performance on correctness & error QA. In addition, we also evaluate a setup where we ask models to generate their own descriptions of students’ responses, and provide those captions in place of gold ones.

7.2 Results

Results are in the expected direction, in that VLM performance on correctness & errors QA improves

⁷We remove 262 images out of a total of 2,030. We considered editing captions rather than remove images, but correctness-related content was sometimes integrated with other caption content and would require intensive rewriting.

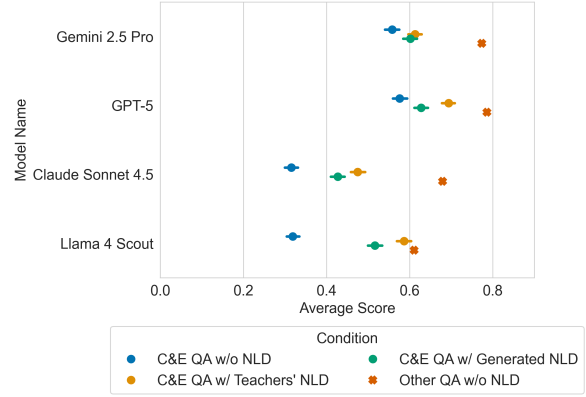


Figure 7: Model performance on correctness & error (C&E) QA, with and without natural language description (NLD) support. We evaluate with a subset of input images and captions as described in §7.1. Error bars are 95% CI.

with natural language support (Figure 7). However, this improved performance on correctness & errors QA with captions still lags behind VLMs’ caption-less performance in all other question categories for the same set of images. So, it is challenging for VLMs to make higher-level inferences useful for pedagogy even with gold textual support. A fully automatic two step caption-then-answer-QA process is a form of test-time scaling. We find that providing models their own generated captions of images can get models close to, but not match, their performance with teacher-written gold captions (Figure 7).

8 Binary judgements of student correctness remain challenging

Open-ended questions are inherently more difficult than binary (e.g. yes/no) questions, as the latter is more guessable. Correctness & errors QA (**F2**) in DrawEduMath provide a natural testbed for comparing open-ended questions (“What errors does the student make in their response?”) and binary questions that assess whether some aspect of a student’s response is correct/incorrect.

8.1 Analysis Setup

Our analysis splits correctness & error QA into the following three subcategories:

- Generic questions (45.0%). This is the open-ended question that DrawEduMath includes for all student images: “What errors does the student make in their response?”

- Binary assessments of specific solution components (50.4%), e.g. “Does the student put the decimal in the correct place in the product?”
- Other questions (4.5%), which mostly pertain to the nature of a student’s error, e.g. “What incorrect product did the student calculate for 667 times 5?”

We use GPT-5-mini as an annotator to label whether non-generic questions are binary or other (prompt in Appendix B.2). We validate this LM annotator on a manually labeled random sample of 200 unique questions ($F1 = 0.975$, Cohen’s $\kappa = 0.934$). We focus on binary and generic in the main text; Appendix D includes some results involving other.

Do models tend to predict that students make errors when they don’t, or do they tend to overlook errors instead? For binary QA, we use GPT-5-mini to annotate whether questions and gold answers indicate that student is correct or incorrect (prompt in Appendix B.3). For example, for the ground truth binary QA pair “*Q: Is there an error in the way that the number line has been drawn? A: Yes*”, the LM annotator would output that the student is *incorrect*. We validate this LM annotator on a sample of 200 manually annotated random examples ($F1 = 0.911$, Cohen’s $\kappa = 0.854$). In total, the LM annotator labels 59.01% of 2,274 binary QA as cases where the specified aspect of the student’s response is correct, and the rest as ones where the specified aspect is incorrect.

We also examine model performance on generic QA, disaggregated by whether the student’s response is overall incorrect or correct. In DrawEduMath, successfully answering this question for correct students requires simply stating that there is no error, while for models to score well for erroneous students, they must also faithfully describe error specifics.

8.2 Results

Figure 8 shows that performance patterns on correctness & error QA, in relation to student correctness, generally and specifically, tend to be idiosyncratic. Some models tend to overreport errors being present, with lower scores on student images with *no error*. Others struggle to detect and, in the case of generic, articulate errors that are present, with lower scores on student images with *error*. Model behavior patterns are not shared

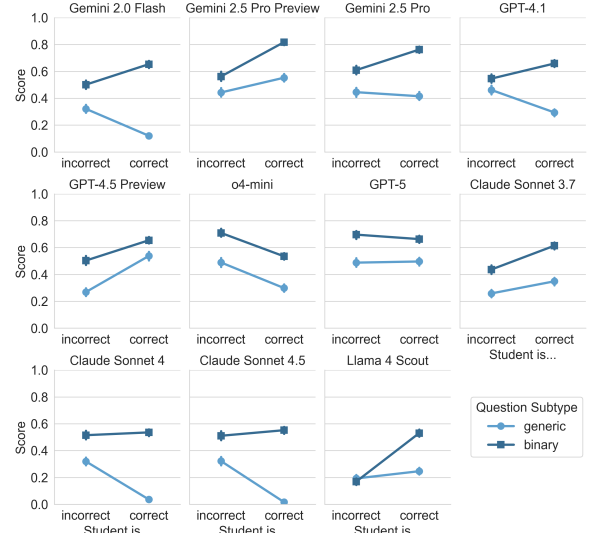


Figure 8: VLMs’ performance on the two main subtypes of correctness & error QA, disaggregated by whether a student response is overall correct (generic) or correct based on specific aspect of their solution (binary). Error bars are 95% CI.

across model versions from the same family or developer. Figure 8 also indicates that some VLMs’ binary QA scores hover closely around a random baseline of 0.5. Overall, assessing student error is incredibly challenging for VLMs, even though a substantial proportion of correctness & error QA in DrawEduMath have high by-chance floor for performance.

9 Conclusion

Despite increasing attention towards the use of multimodal AI in education, our evaluation of 11 models released in 2025 demonstrates that their application on real student data remains challenging. Our findings suggest that erroneous student work is inherently more difficult for VLMs than correct student work (F1). VLM training and evaluation pipelines that favor correct mathematical content are at tension with the promise of AI for education, where incorrect math requires extra emphasis and attention. We also show that QA involving assessments of student correctness are particularly tricky (F2), across both text and image inputs, and across open-ended and binary question forms.

Though this present paper presents a detailed error analysis of VLMs’ performance on one vision-language K-12 math benchmark, our evaluation approach can be re-applied to other education-related benchmarks as well. That is, the evaluation of AI in education should be disaggregated in a man-

ner that pinpoints whether models can actually discern when a student may need pedagogical support (F2), and whether they equitably serve students across different levels of proficiency (F1). Without a careful eye on the latter, models’ capabilities may be overstated, and rushed integration into classrooms may exacerbate existing academic achievement gaps.

Limitations

Scope and data representativeness. Our study focuses on a single English benchmark, which involves student response images drawn from one online learning platform, ASSISTments. Thus, our findings may not map directly onto other languages and learning contexts. Based on school-level data provided by ASSISTments, we estimate that 85% of images come from Title I schools, which are public schools in the U.S. that receive federal funding to support low-income students. ASSISTments partners with teachers and schools located across multiple location types (e.g. rural, suburban, town, city) and regions (e.g. West Coast, Midwest, East Coast, South), but self-selection is at play when it comes to which teachers, schools, and districts use the platform. DrawEduMath also contains questions that represent what was salient to Teaching Lab’s teacher annotators (Baral et al., 2025); it is not comprehensive of all of the ways in which educators may interpret and support student learning. Still, our high-level evaluation approach can be re-applied to other benchmarks and contexts, because transparency around the impact of student error on model performance is relevant to nearly all education-related settings.

Data and evaluation granularity. We examine performance on DrawEduMath on aggregate with a binary scoring approach, both for students’ correctness and LMs’ VQA correctness. We acknowledge that the evaluation of VLMs may benefit from a more nuanced approach that captures multiple constructs beyond judgments of correct/incorrect. For example, future work should consider the form of student error present (e.g. arithmetic versus conceptual error). Future work should also consider more sophisticated evaluation frameworks that consider the context and complexity of different math problems, and the evaluative goals of different QA items (e.g. QA that pertain to VLMs’ descriptive abilities, versus those that pertain to pedagogical abilities).

Data constraints. Some of our experiments and analyses navigate practical, data-related constraints. For example, our image redrawing experiment in §5 uses a small sample rather than the full dataset, since redrawing is a time-intensive process. Our other analyses rely on pre-existing teacher annotations and data present in DrawEduMath. For example, in §7, we removed some images from our analysis because their captions contained correctness & error information, because models’ performance with textual support on those images would be inflated. In addition, the content description QA under consideration for the results shown in Table 3 are only questions shared across multiple student images for a problem, for which we could gather sufficient signal for what correct student response behavior should be. So, our results in that section (§6) primarily serve to illustrate one possible explanation for models’ performance, and is not comprehensive of all of DrawEduMath.

Ethical Considerations

Education is a high-stakes setting for VLM use and deployment. The intermixing of AI and education involves delegating pedagogy to automated systems, impacting vulnerable underage populations, with possible life-long downstream effects related to economic mobility. Though there is optimism around AI’s ability to support education (Demszky et al., 2025), there should also be caution that it does not exacerbate existing inequities or introduce new ones (Winters et al., 2020; Harvey et al., 2024). We acknowledge that our work focuses primarily on technical harms measurable from model outputs, and does not capture broader harms that may emerge via interaction of AI with students, teachers, and school systems (Harvey et al., 2025). In addition, AI research often involves a deployment-first mentality, where deployment may occur before a system has been deemed functional or necessary (Raji et al., 2022). Our work advocates for robust evaluation and auditing of AI prior to deployment (Raji et al., 2020), and accountability behind claims around model functionality and its social benefits (Kou et al., 2025; Wang et al., 2024).

Acknowledgments

We are grateful for valuable feedback from Douglas Jaffe, who encouraged us to dig further into the impact of student error on model performance. We are also grateful for data-related support from Sami

Baral, Neil Heffernan, and Cristina Heffernan. Our work is funded by the Gates Foundation.

References

- Shaden Alshammari, Kevin Wen, Abrar Zainal, Mark Hamilton, Navid Safaei, Sultan Albarakati, William T. Freeman, and Antonio Torralba. 2026. [MathNet: A global multimodal benchmark for mathematical reasoning and retrieval](#). In *The Fourteenth International Conference on Learning Representations*.
- Lidia Altamura, Cristina Vargas, and Ladislao Salmerón. 2025. Do new forms of reading pay off? A meta-analysis on the relationship between leisure digital reading habits and text comprehension. *Review of Educational Research*, 95(1):53–88.
- Lisa Anthony, Jie Yang, and Kenneth R. Koedinger. 2007. Benefits of handwritten input for students learning algebra equation solving. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, page 521–523, NLD. IOS Press.
- Sami Baral, Li Lucy, Ryan Knight, Alice Ng, Luca Soldaini, Neil Heffernan, and Kyle Lo. 2024. [Drawedumath: An expert-annotated dataset of students’ math images](#). In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS’24*.
- Sami Baral, Li Lucy, Ryan Knight, Alice Ng, Luca Soldaini, Neil Heffernan, and Kyle Lo. 2025. [DrawEduMath: Evaluating vision language models with expert-annotated students’ hand-drawn math images](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6902–6920, Albuquerque, New Mexico. Association for Computational Linguistics.
- Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. 2025. [Generative AI without guardrails can harm learning: Evidence from high school mathematics](#). *Proceedings of the National Academy of Sciences*, 122(26):e2422633122.
- Su Lin Blodgett and Michael Madaio. 2021. Risks of AI foundation models in education. *arXiv preprint arXiv:2110.10024*.
- Raffaella Borasi. 1994. [Capitalizing on errors as “springboards for inquiry”: A teaching experiment](#). *Journal for Research in Mathematics Education*, 25(2):166–208.
- Anthony Botelho, Sami Baral, John A. Erickson, Priyanka Benachamardi, and Neil T. Heffernan. 2023. [Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics](#). *Journal of Computer Assisted Learning*, 39(3):823–840.
- Valerio Capraro, Austin Lentsch, Daron Acemoglu, Selin Akgun, Aisel Akhmedova, Ennio Bilancini, Jean-François Bonnefon, Pablo Brañas-Garza, Luigi Butera, Karen M Douglas, and 1 others. 2024. The impact of generative artificial intelligence on socioeconomic inequalities and policy making. *PNAS nexus*, 3(6):pgae191.
- Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. [POSIX: A prompt sensitivity index for large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Josh Cows, Andreas Tsamados, Mariarosaria Taddeo, and Luciano Floridi. 2021. A definition, benchmark and database of AI for social good initiatives. *Nature Machine Intelligence*, 3(2):111–115.
- Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. [Stepwise verification and remediation of student reasoning errors with large language model tutors](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.
- Dorottya Demszky, Jing Liu, Heather C. Hill, Shyamoli Sanghi, and Ariel Chung. 2025. [Automated feedback improves teachers’ questioning quality in brick-and-mortar classrooms: Opportunities for further enhancement](#). *Computers & Education*, 227:105183.
- Mingyu Feng, Linlin Li, Chunwei Huang, Natalie Brezack, and Kim Luttgen. 2025. Empowering teachers with technology: A national study on a formative assessment platform. In *International Conference on Artificial Intelligence in Education*, pages 119–126. Springer.
- Esther Gan, Yiran Zhao, Liying Cheng, Mao Yancan, Anirudh Goyal, Kenji Kawaguchi, Min-Yen Kan, and Michael Shieh. 2024. [Reasoning robustness of LLMs to adversarial typographical errors](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10449–10459, Miami, Florida, USA. Association for Computational Linguistics.
- Google DeepMind. 2025. [Gemini 3 Pro model card](#).
- Dorit Hadar Shoval. 2025. [Artificial intelligence in higher education: Bridging or widening the gap for diverse student populations?](#) *Education Sciences*, 15(5).

- Emma Harvey, Allison Koenecke, and Rene F Kizilcec. 2024. Towards an educator-centered method for measuring bias in large language model-based chatbot tutors. In *AI for Education: Bridging Innovation and Responsibility at the 38th AAAI Annual Conference on AI*.
- Emma Harvey, Allison Koenecke, and Rene F. Kizilcec. 2025. "Don't forget the teachers": Towards an educator-centered understanding of harms from large language models in education. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.
- Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Kenneth Holstein and Shayan Doroudi. 2021. Equity and artificial intelligence in education: Will "AIED" amplify or alleviate inequities in education? *Preprint*, arXiv:2104.12920.
- Hyoungwook Jin, Yoonsu Kim, Dongyun Jung, Seungju Kim, Kiyeon Choi, Jinho Son, and Juho Kim. 2025. Investigating large language models in diagnosing students' cognitive skills in math problem-solving. *arXiv preprint arXiv:2504.00843*.
- Rogers Kaliisa, Kamila Misiejuk, Sonsoles López-Pernas, and Mohammed Saqr. 2026. How does artificial intelligence compare to human feedback? a meta-analysis of performance, feedback perception, and learning dispositions. *Educational Psychology*, 46(1):80–111.
- Blanka Klimova and Marcel Pikhart. 2025. Exploring the effects of artificial intelligence on student and academic well-being in higher education: A mini-review. *Frontiers in Psychology*, 16:1498132.
- Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1011–1033, Vienna, Austria. Association for Computational Linguistics.
- Tianqi Kou, Dana Calacci, and Cindy Lin. 2025. Dead zone of accountability: Why social claims in machine learning research should be articulated and defended. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 1501–1512.
- Stefan Küchemann, Karina E Avila, Yavuz Dinc, Chiara Hortmann, Natalia Revenga, Verena Ruf, Niklas Stausberg, Steffen Steinert, Frank Fischer, Martin Fischer, and 1 others. 2025. On opportunities and challenges of large multimodal foundation models in education. *npj Science of Learning*, 10(1):11.
- Ehsan Latif, Zirak Khan, and Xiaoming Zhai. 2025. SketchMind: A multi-agent cognitive framework for assessing student-drawn scientific sketches. *arXiv preprint arXiv:2507.22904*.
- Hao-Ping (Hank) Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. 2025a. The impact of generative ai on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.
- Jimin Lee, Steven-Shine Chen, and Paul Pu Liang. 2025b. Interactive Sketchpad: A multimodal tutoring system for collaborative, visual problem-solving. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '25, New York, NY, USA. Association for Computing Machinery.
- Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*.
- Rabeeh Karimi Mahabadi, Sanjeev Satheesh, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. Nemotron-CC-Math: A 133 billion-token-scale high quality math pretraining dataset. *Preprint*, arXiv:2508.15096.
- Janet Metcalfe. 2017. Learning from errors. *Annual Review of Psychology*, 68:465–489.
- Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. Towards automated document revision: Grammatical error correction, fluency edits, and beyond. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications*

- (BEA 2024), pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.
- Pam A. Mueller and Daniel M. Oppenheimer. 2014. [The pen is mightier than the keyboard: Advantages of longhand over laptop note taking](#). *Psychological Science*, 25(6):1159–1168. PMID: 24760141.
- Oikantik Nath, Hanani Bathina, Mohammed Safi Ur Rahman Khan, and Mitesh M Khapra. 2025. [Can vision-language models evaluate handwritten math?](#) In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14784–14814, Vienna, Austria. Association for Computational Linguistics.
- Thu Phuong Nguyen, Duc M. Nguyen, Hyotaek Jeon, Hyunwook Lee, Hyunmin Song, Sungahn Ko, and Taehwan Kim. 2025. [VEHME: A vision-language model for evaluating handwritten mathematics expressions](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31793–31813, Suzhou, China. Association for Computational Linguistics.
- Behnam Parsaeifard, Martin Hlosta, and Per Bergamin. 2025. Automated grading of students’ handwritten graphs: A comparison of meta-learning and vision-large language models. *arXiv preprint arXiv:2507.03056*.
- Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. [OpenWebMath: An open dataset of high-quality mathematical web text](#). In *The Twelfth International Conference on Learning Representations*.
- Hendrik Radatz. 1979. Error analysis in mathematics education. *Journal for Research in mathematics Education*, 10(3):163–172.
- Inioluwa Deborah Raji, I Elizabeth Kumar, Aaron Horowitz, and Andrew Selbst. 2022. The fallacy of AI functionality. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 959–972.
- Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. [Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing](#). In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* ’20, page 33–44, New York, NY, USA. Association for Computing Machinery.
- Bethany Rittle-Johnson, Rebecca Adler, Kelley Durkin, L Burleigh, Jules King, and Scott Crossley. 2025. [Detecting math misconceptions: An AI benchmark dataset](#). In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Works in Progress*, pages 20–24, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- Alexis Ross and Jacob Andreas. 2025. [Learning to make MISTAKES: Modeling incorrect student thinking and key errors](#). *Preprint*, arXiv:2510.11502.
- Nils-Jonathan Schaller, Yuning Ding, Andrea Horbach, Jennifer Meyer, and Thorben Jansen. 2024. [Fairness in automated essay scoring: A comparative analysis of algorithms on German learner essays from secondary education](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 210–221, Mexico City, Mexico. Association for Computational Linguistics.
- Judy Hanwen Shen and Alex Tamkin. 2026. How AI impacts skill formation. *arXiv preprint arXiv:2601.20245*.
- John P Smith III, Andrea A. diSessa, and Jeremy Roschelle. 1994. [Misconceptions reconceived: A constructivist analysis of knowledge in transition](#). *Journal of the Learning Sciences*, 3(2):115–163.
- Neha Srikanth, Rupak Sarkar, Heran Mane, Elizabeth Aparicio, Quynh Nguyen, Rachel Rudinger, and Jordan Boyd-Graber. 2024. [Pregnant questions: The importance of pragmatic awareness in maternal health question answering](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7253–7268, Mexico City, Mexico. Association for Computational Linguistics.
- Kv Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2025. [LLMs cannot spot math errors, even when allowed to peek into the solution](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10914–10928, Suzhou, China. Association for Computational Linguistics.
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM prompting strategies for joint essay scoring and feedback generation](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.
- Keita Umejima, Takuya Ibaraki, Takahiro Yamazaki, and Kuniyoshi L Sakai. 2021. Paper notebooks vs. mobile devices: Brain activation differences during memory retrieval. *Frontiers in Behavioral Neuroscience*, 15:634158.
- Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. 2024. [Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy](#). *ACM J. Responsib. Comput.*, 1(1).
- Ryan Yixiang Wang, Matthew Finlayson, Luca Soldaini, Swabha Swayamdipta, and Robin Jia. 2025. [Teaching models to understand \(but not generate\) high-risk data](#). In *Second Conference on Language Modeling*.

Niall Winters, Rebecca Eynon, Anne Geniets, James Robson, and Ken Kahn. 2020. [Can we avoid digital structural violence in future learning systems?](#) *Learning, Media and Technology*, 45(1):17–30.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2025. [A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11798–11827, Vienna, Austria. Association for Computational Linguistics.

Xinyan Yu, Sewon Min, Luke Zettlemoyer, and Han-naneh Hajishirzi. 2023. [CREPE: Open-domain question answering with false presuppositions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10457–10480, Toronto, Canada. Association for Computational Linguistics.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. 2024. [MATHVERSE: Does your multi-modal LLM truly see the diagrams in visual math problems?](#) In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VIII*, page 169–186, Berlin, Heidelberg. Springer-Verlag.

A Model Performance by Question Type

In the main text, Figure 3 shows that the gap in VLM performance between erroneous and non-erroneous student responses is primarily driven by content description QA. Figure 10 expands upon that finding, by disaggregating content description QA’s overall pattern into finer-grained question categories and showing results for all 11 VLMs. Across these expanded plots, we see that the performance gap between erroneous and non-erroneous student responses persists across finer-grained content description QA categories.

B Language Model-Assisted Data Annotation

For each LM-assisted labeling task, we iteratively developed prompts that yield solid performance on small samples, before validating our final prompts on larger samples. The main text details the performance of each prompt on the intended task.

B.1 Student Error

One of our main findings, **F1**, pertains to how models perform differently between student responses that contain errors versus those that do not. To

determine whether a student response contains an error or not, we rely on teachers’ free-form descriptions of student error. Since teachers’ written responses may span a variety of phrasings, we use GPT-5-mini to decisively label whether the teacher indicates that the student response contains an error. Here, *ans* is the teacher’s answer to the question, *What errors does the student make in their response?* We use the following prompt:

When asked about what errors a student makes in their response to a math problem, a teacher writes, '{ans}'. Based on the teacher's feedback, does the student make any error? Respond 'yes' or 'no'.

B.2 Finer-grained Correctness & Error Questions

In §8.1, we discuss how correctness & error QA span both binary assessments of specific student errors and more-open-ended questions. We use GPT-5-mini as an annotator to label whether each Correctness & Error question is binary or other. We use the following prompt:

Is the following question a binary question that asks whether a student does something correctly or not?
Question: '{question}'
Decide whether the question above is a binary question that judges a student's correctness. Your response should start with 'Yes' or 'No':

B.3 Binary Student Correctness

In §8.2, we’re interested in investigating whether models tend to under- or over-report student error on binary correctness & error questions. We use GPT-5-mini to annotate whether binary QA’s questions and gold answers indicate that student is correct or incorrect. We use the following prompt:

Teacher A is examining a student's solution to a math problem. Teacher B asks Teacher A, '{question}'
Teacher A says, '{answer}'.
Does this exchange indicate that the student's solution has an error? Respond "yes" or "no":

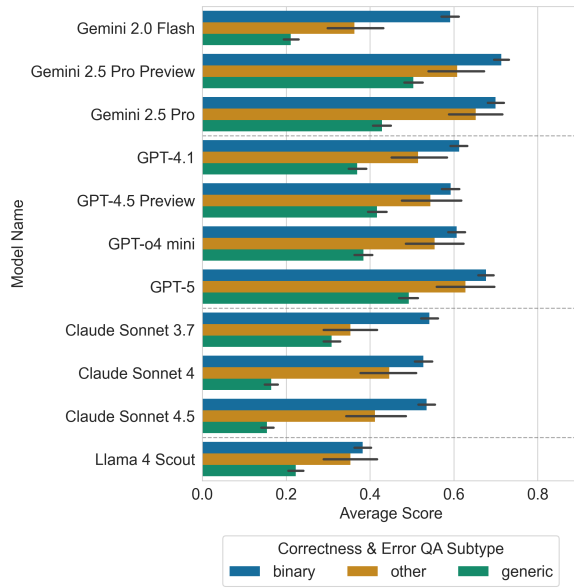


Figure 9: VLMs’ performance across different subtypes of correctness & error questions, as defined in §8.1.

C Natural Language Description Experiments

In §7, we investigate whether the inclusion of textual descriptions of students’ work can support VLMs’ abilities to make higher-level inferences around students’ correctness and errors.

C.1 Prompts

Our prompt that adds in natural language descriptions/captions is intuitive and simple:

```
Description of image:
{caption}

Answer the following question: {question}
```

To generate descriptions or captions using language models, we use the following prompt: Describe the Student Response on the right side of the image in one paragraph.

D Additional Correctness & Error Results

The relative performance ranking of binary, other, and generic correctness & error QA is consistent across all 11 VLMs (Figure 9).

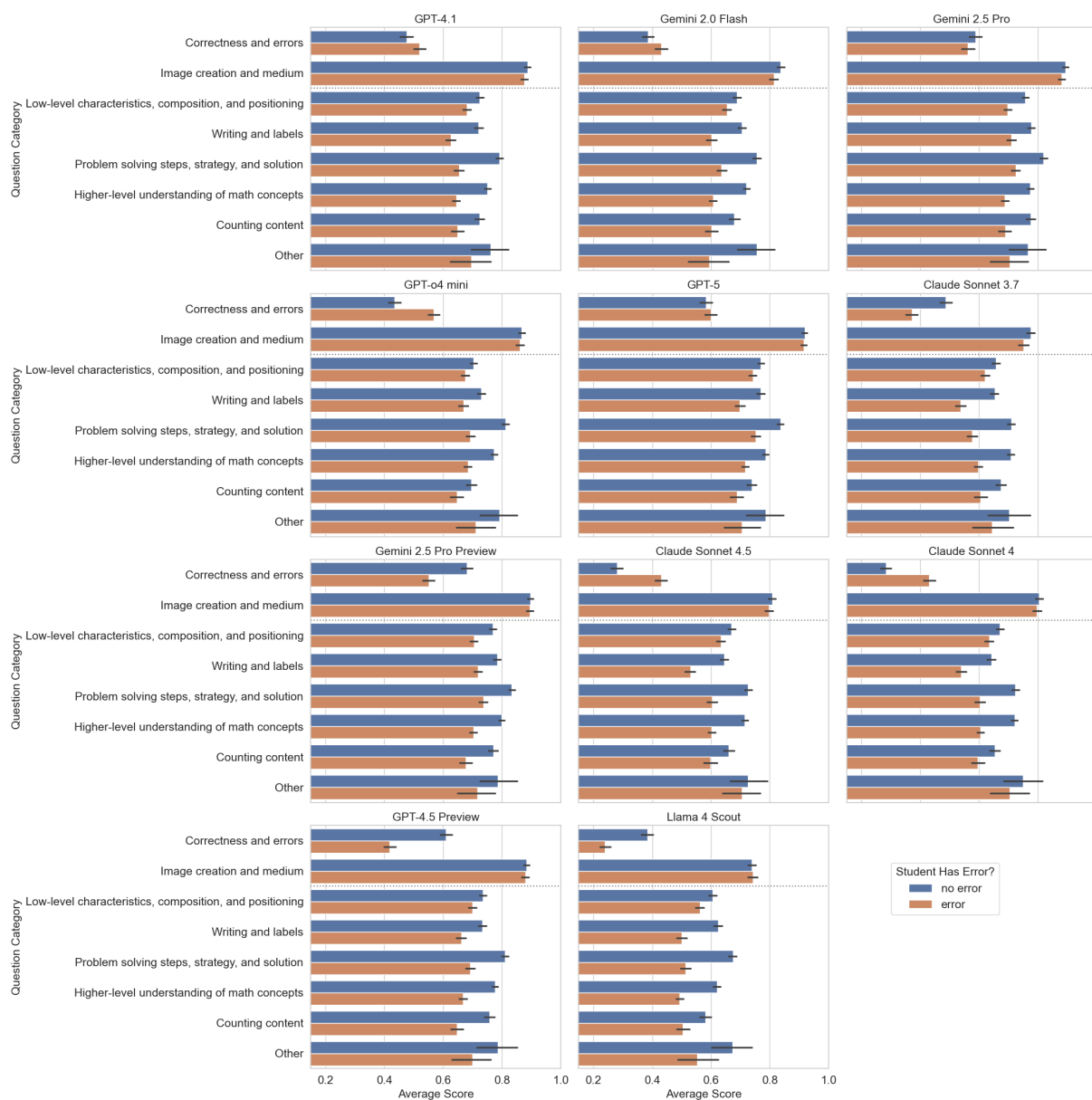


Figure 10: An expanded version of Figure 3, showing which question categories contribute to the gap in VLM performance between student responses that contain errors versus those that do not. Questions below the dotted line in each subplot are content description QA.

Criterial Features in German: Towards Interpretable NLP in Readability Assessment

Denise Löfflad¹, Sofia Kathmann¹, Heiko Holz², Detmar Meurers^{1,3}

¹Leibniz-Institut für Wissensmedien Tübingen, Germany

²University of Education Ludwigsburg, Germany ³University of Tübingen, Germany,

d.loefflad@iwm-tuebingen.de, s.kathmann@iwm-tuebingen.de

heiko.holz@ph-ludwigsburg.de, d.meurers@iwm-tuebingen.de

Abstract

This paper presents an empirical evaluation of the German Grammar Profile (GGP), a CEFR-aligned resource of criterial features, and its corresponding extraction system PALME.

We design a systematic test suite in which each feature extractor is evaluated on controlled positive and negative examples. The results show that PALME achieves high precision and recall across all CEFR levels, with over 90% of features achieving scores above 0.8. Qualitative analysis shows that lower performance primarily results from morphological ambiguity in noun and adjective case marking.

To evaluate the usefulness of the criterial features of the GGP for CEFR-aligned readability assessment, we assess their predictive power using Explainable Boosting Machines on graded readers. The model achieves strong performance (precision: 0.75, recall: 0.73). Our qualitative analysis shows that features related to specific verb constructions follow patterns consistent with developmental stages predicted by Processability Theory. These findings underline the value and relevance of criterial features for modeling language development in readability assessment.

1 Introduction

The quality of available language input plays an important role in second language acquisition (SLA), with reading serving as a crucial source of input. Automatic Readability Assessment (ARA) aims to predict the readability of a text and can support teachers in selecting appropriate texts more efficiently. Different approaches to ARA have been implemented in the past, including approaches based on surface-based readability formulas (Collins-Thompson, 2014), or using machine learning methods based on aggregated language characteristics (Weiss and Meurers, 2021; Vázquez-Ingelmo et al., 2026). These approaches typically assign a level to

an entire text, allowing for a holistic view of readability. However, while such methods often achieve strong predictive performance, they offer limited insight into the specific language constructions in a text that contribute to its difficulty. This limitation is particularly relevant in language education, where on the one hand input needs to be comprehensible to be effective, but it should also systematically offer developmentally proximal language structures, as pointed out by SLA approaches since the *i+1* Input Hypothesis of Krashen (1985) and Pienemann’s Teachability Hypothesis (Pienemann, 1989, 2015).

Criterial features (CFs; Hawkins and Buttery, 2010; Hawkins and Filipović, 2012) have been proposed as more linguistically and pedagogically grounded measures for proficiency and ARA. They play a central role in operationalizing the Common European Framework of Reference for Languages (CEFR; Council of Europe, 2020). As CEFR descriptors define purely communicative competences in the form of language-agnostic Can-Do statements, they deliberately refrain from specifying concrete linguistic realizations. Reference level descriptors aim to capture links between linguistic properties of texts or learner production and CEFR levels, and CFs are linguistic properties that are characteristic of specific stages of proficiency. CFs can thus be understood as an attempt to characterize the communicative purposes of CEFR descriptors linguistically, making them measurable and analyzable using computational linguistic methods.

For English, the English Grammar Profile (EGP, O’Keeffe and Mark, 2017) offers a resource of CFs that link linguistic aspects to CEFR levels. The EGP has been used as starting point to build computer-based systems to automatically extract those features (Sagirov and Chen, 2025). For German, comparable resources such as the German Grammar Profile (GGP) and the corresponding extraction system PALME exist, but they have not yet

been systematically evaluated or empirically validated (Löfflad et al., 2025). However, the lack of systematic evaluation leaves open questions about both the reliability of feature extraction and the validity of the underlying features. This raises two key questions: How can such systems be evaluated, and how can the underlying features be validated?

In this paper, we address these gaps by proposing (i) systematic methods for evaluating and validating CF-based system from both technological and pedagogical perspectives, (ii) a dedicated resource for evaluating German CF extraction systems, including potential applications for Large Language Models (LLMs), and (iii) a contribution to the theoretical understanding of CFs through a comparison with developmental trajectories (Pienemann, 1998). More broadly, we aim to provide a framework that can be transferred to other languages, supporting the extension of CEFR- and CF-related resources.

We investigate the computational operationalizability and empirical validity of CFs as indicators of CEFR readability in German graded readers and ask two research questions:

- RQ1 How can criterial features reliably be operationalized and automatically extracted?
- RQ2 To what extent do automatically detected criterial features predict CEFR levels and link to developmental trajectories?

To address the first question, we construct a systematic test suite of example sentences to evaluate the automatic extraction system PALME. For the second question, we use PALME to extract all 153 CFs in the GGP from a graded reader corpus and train an Explainable Boosting Machine (EBM) to predict readability levels. The EBM further enables us to analyze developmental trajectories and examine how CFs evolve across readability levels.

2 Background

2.1 Criterial Features and Developmental Trajectories

Criterial features reflect linguistic properties that are reliably acquired at specific stages of second language development and can thus serve as indicators of proficiency (Hawkins and Buttery, 2010) and are typically aligned with the CEFR. Recent research has increasingly turned to criterial features as a concrete, developmentally informed and transparent possibility to measure linguistic phenomena (Salamoura and Saville, 2010; Hawkins

and Filipović, 2012; Gaillat et al., 2022). The goal of criterial feature research is to identify the linguistic phenomena that formalize underspecified descriptors as part of a more systematic resource.

Developmental trajectories (also stages or themed sequences) assume that certain grammatical aspects are acquired in predictable sequences. One influential theoretical framework, the Processability Theory (PT), proposes that the acquisition of grammatical structures follows a hierarchical order (Pienemann, 1998). Although criterial feature research aims to capture developmental trajectories, the relationship between trajectories and CEFR descriptors is still under-researched. Previous studies have analyzed the acquisition trajectories of individual grammatical constructs in detail (e.g., Wisniewski, 2020). Importantly, developmental trajectories are primarily based on learner language production. Nevertheless, they should also be examined in reading input such as graded reading materials, which are designed to slightly exceed the learner’s current competence (i+1, Krashen, 1985). Since graded readers are pedagogically designed to align with learner proficiency levels, the distribution of criterial features in such texts should, in principle, reflect the developmental stages associated with those levels.

The EGP offers a well-established resource of criterial features for English (O’Keeffe and Mark, 2017), expanding on CEFR descriptors by mapping them onto fine-grained grammatical structures. There is work for other languages to develop reference level descriptors (Mohamed, 2023; Cuberos Vicente and De Cock, 2023), however, the EGP is a unique resource of language-specific grammatical features. Nevertheless, a similar yet smaller resource for German was developed by Löfflad et al. (2025) as described in the following.

2.2 The German Grammar Profile

The GGP (Löfflad et al., 2025) is a resource of German criterial features developed on the basis of the English Grammar Profile (EGP, O’Keeffe and Mark, 2017) and the *Profil Deutsch* (Glaboniat, 2010), a functional grammar for German as a second language learners and teachers. Following a theory-driven approach, the GGP defines a set of 153 criterial features mapped to CEFR levels based on the functional grammar provided by Glaboniat (2010), and not derived from corpus data (example features can be seen in Appendix A). Of these 153 features, 47 are assigned to level A1, 49 to

level A2, 35 to level B1, and 22 to level B2. The distribution of features across levels is somewhat imbalanced. This is partly due to fewer new grammatical features being introduced at higher levels and partly because higher level features tend to be usage-based, which is not in the scope of the GGP. Nevertheless, this can impact subsequent data analyses, as A level features might be more prominent.

Moreover, the theory-driven approach used to create the GGP, in contrast to the data-driven approach used in the development of the EGP, may lead to mismatches between the levels assigned in the GGP and the levels observed in graded readers or learner data. Such mismatches are plausible given that previous research on the EGP has already found limited agreement between statistically derived feature mappings and the mappings proposed by the EGP (Verratti-Souto et al., 2025).

2.3 The PALME System

To automatically extract the measures defined in the GGP, Löfflad et al. (2025) developed the PALME system, the *Pedagogically Oriented Linguistic Feature Extraction* (*Pädagogisch Ausgerichtete Linguistische Merkmalsextraktion*). PALME relies on rule-based NLP components. While designing such components is labor-intensive, it offers advantages in comparison to Generative AI in terms of reliability and transparency. Once the system has been systematically tested, its behavior is predictable and performance does not depend on changes in black-box AI models.

The system is related to other approaches supporting automatic grammatical feature extraction and annotation. The POLKE system (*Pedagogically Oriented Language Knowledge Extractor*, Sa-girov and Chen, 2025), for example, extracts more than 600 measures from the EGP using a comparable rule-based framework that includes Java and RUTA-based preprocessing. POLKE primarily focuses on the large-scale implementation and evaluation of EGP-based measures. For Portuguese, the SABER system (*Sistema de Análise e Busca de Estruturas Relevantes*, Akef et al., 2025) provides a related approach to grammatical feature extraction.

2.4 Test Suite vs. Corpus-Based Evaluation

Test suite-based evaluations inherently differ from corpus-based evaluations: While corpus-based evaluations assess systems on naturally occurring language data, test suites consist of carefully constructed, focused examples of language use

(Lehmann et al., 1996; Prasad and Sarkar, 2000). Both methods come with advantages and disadvantages. Corpus-based evaluations facilitate testing a system’s robustness against authentic language and are well-suited for frequent linguistic constructs. However, many linguistic constructs occur only rarely in natural data. Test suites address this limitation by supporting a controlled evaluation that systematically evaluates the coverage of specific constructs (Lehmann et al., 1996).

For a similar extraction system for English, Sa-girov and Chen (2025) used a corpus-based evaluation approach. They randomly selected 15 features for each CEFR level and automatically extracted sentences containing the respective features from the Corpus of Contemporary American English (Davies, 2009). In addition, they randomly selected sentences that did not contain the feature as negative examples.

3 Methods

3.1 System Evaluation

To evaluate the system and answer RQ1, we opted for a test suite-based approach. This allowed us to control the evaluation sentences, particularly the negative examples, enabling us to construct challenging cases that stress-test the system. Moreover, our goal was to evaluate every measure extracted by PALME. As mentioned previously, some features occur only rarely in natural language data, which makes them difficult to capture reliably through corpus-based sampling. Although there are, to the best of our knowledge, no widely accepted guidelines on writing test suites, Balkan et al. (1994) recommend to include both positive and negative examples of the target feature. They further suggest covering a range of instantiations and to intentionally add misleading inputs and traps for the system. We thus constructed a systematic test suite consisting of positive and negative examples for each feature in the GGP. For every feature, the test suite contains ten sentences: five positive examples that include the target feature and five negative examples that do not.

To ensure variability and linguistic diversity, the sentences were developed collaboratively by three experts: the first author, the second author as a natural language processing (NLP) expert, and a pedagogical expert in second language teaching. The positive examples were informed by reading and learner corpora, but were adapted or constructed

where necessary to ensure controlled evaluation, for example by avoiding multiple instances of the same feature or by including theoretically motivated constructions. We deliberately included different structural and lexical realizations of each feature to capture its range of possible realizations. For the negative examples, we constructed sentences that were closely related to the target feature, either structurally, linguistically, or pedagogically, but did not actually instantiate it. For example, in the case of adjectival case marking without an article, accusative constructions were contrasted with structurally parallel sentences in other cases. Similarly, for verbal features, negative examples were designed to resemble the relevant constructions while differing in tense, agreement, or syntactic realization. This design ensured that the evaluation tested the system’s ability to distinguish subtle contrasts. After constructing the test suite, we evaluated the system’s performance on all features. We computed precision and recall for each feature and identified those with scores below a threshold of 0.8. As each feature is represented by five positive and five negative examples, a score of 0.8 or higher corresponds to at least four correctly classified instances within a set of five (i.e., at most one error). These features were subsequently revised and improved, and the sentences targeting the revised features were updated with new examples to prevent overfitting to the original test sentences. Following each revision cycle, the system was re-evaluated on the test suite. Features that continued to fall below the performance threshold were further refined. This iterative process was conducted for two improvement cycles and concluded with a final evaluation. Although additional cycles might have led to further performance gains, we stopped after the third evaluation round to minimize the risk of overfitting the system to the test suite.

3.2 Feature Validation

After completing the system optimization phase, we validated the extracted CFs on graded reader data to answer RQ2. We used texts provided by *Deutsch Perfekt*¹, a magazine that publishes reading materials for second language learners.

While this dataset does not consist of learner production, graded reading materials provide an approximation of proficiency levels and are therefore a suitable starting point for evaluating the usefulness

of criterial features in readability classification. Given that this study represents an initial step toward a broader validation of CFs, we focus here on input texts, which play a central role in language development. In line with pedagogical principles such as the *i+1* hypothesis (Krashen, 1985), reading materials should be designed to be just above the learner’s current proficiency level, thereby reflecting increasing linguistic complexity across levels.

The corpus is categorized into three proficiency levels: easy (approximately CEFR A2), intermediate (approximately B1), and advanced (approximately B2/C1). In total, the dataset consists of 4,906 texts spanning these proficiency levels. Table 1 shows a descriptive overview of the corpus.

Level	Num. Texts	Num. Tokens
Easy	2,535	447,287
Intermediate	1,463	830,158
Advanced	908	461,446
Total	4,906	1,738,891

Table 1: Overview of the Deutsch Perfekt corpus

The classes are imbalanced, with most tokens in the corpus being part of texts in the intermediate class. We conducted a predictive evaluation by training an Explainable Boosting Machine (EBM) on the extracted CFs to predict CEFR levels using an 80–20 train–test split. We normalized raw feature counts by calculating feature occurrences per 1,000 tokens to control for differences in text length. EBMs are generalized additive models that learn each feature function using gradient boosting and maintaining additive structures, thus enabling high interpretability (Nori et al., 2019; Lou et al., 2013). EBMs train each feature function one at a time, allowing for each feature function to be examined individually. They belong to so-called "glassbox" models providing transparent feature effects, in contrast to post-hoc explanation methods (e.g., LSTMs or SVMs). Using EBMs instead of other machine learning models allowed us not only to assess predictive performance but also to interpret the contribution and shape of individual features in relation to proficiency level.

4 Results

4.1 System Evaluation

We first report system performance aggregated of CFs by CEFR level in order to assess overall robustness across proficiency bands. Table 2 presents

¹<https://www.spotlight-verlag.de>

precision, recall, and F1 scores for A1–B2. Performance is consistently high across proficiency levels, with precision ranging from 0.956 (A1) to 0.981 (B1). Recall shows a slight decrease with increasing proficiency, ranging from 0.932 at A1 to 0.855 at B2.

Macro-averaged performance across levels yields a precision of 0.968, recall of 0.886, and an F1 score of 0.925. Micro-averaged performance across all 1,530 instances is similarly high. Our results are in line with those reported by [Sagirov and Chen \(2025\)](#) for the POLKE system, indicating state of the art performance.

Level	Precision	Recall	F1	N
A1	0.956	0.932	0.944	470
A2	0.978	0.894	0.934	490
B1	0.981	0.863	0.918	350
B2	0.959	0.855	0.904	220
Macro Avg.	0.968	0.886	0.925	–
Micro Avg.	0.969	0.893	0.929	1530

Table 2: Precision, recall, and F1 scores per CEFR level. Macro averages are computed across levels; micro averages are computed across all instances.

Across the 153 evaluated measures, 138 measures (90.1%) achieved both precision and recall values of at least 0.8. Table 3 lists the 15 measures (9.9%) that fell below this threshold in at least one of the two metrics. As described in Section 3, the threshold of 0.8 corresponds to at most one error within the five positive or five negative test instances per feature. Table 5 in Appendix A

shows feature descriptions from the GGP and examples sentences in German and English for all features discussed in this paper.

Overall, the results indicate robust and reliable detection performance for the large majority of measures. The measures with lower scores are largely explainable. Several involve case marking (e.g., adjectival inflection without an article or genitive marking), which depends on accurate morphological analysis. We found that the taggers are currently not able to reliably annotate the accusative case, as the form is often similar to the nominative case, which leads to subsequent errors in the rule-based system.

Feature 32148 (subjunctive II) is primarily form-based and involves subtle semantic distinctions that are difficult to operationalize computationally. The comparatively lower recall for *indefinite pronouns* (3168) is less immediately expected and warrants further inspection. Importantly, precision is consistently high across measures, with the large majority reaching a value of 1.0. This indicates that false positives are rare. For subsequent corpus-based analyses, this is particularly desirable, as it minimizes the risk of overestimating the frequency of linguistic structures. A full table of performance for all features can be seen in Appendix B – each feature is assigned a construct ID that encodes the CEFR level: IDs beginning with 21 correspond to A1 features, 22 to A2 features, 31 to B1 features, and 32 to B2 features.

ID	Level	Feature	Precision	Recall
3132	B1	Superlative of B1 level adverbs	1.00	0.20
2264	A2	Adjectives in the accusative after zero articles	1.00	0.40
2268	A2	A2 level Indefinite Pronouns	1.00	0.40
32148	B2	Subjunctive II	1.00	0.40
2111	A1	Negation article ‘kein’ in the accusatives	1.00	0.60
2113	A1	Personal pronouns in the accusative	0.75	0.60
2263	A2	Adjectives in the nominative after zero article	1.00	0.60
2265	A2	Adjectives in the dative after zero article	1.00	0.60
2281	A2	Past Participle	1.00	0.60
3168	B1	B1 level Indefinite pronouns	1.00	0.60
31120	B1	Articles in genitive case	1.00	0.60
3134	B1	Adverbial temporal determinations	1.00	0.60
3256	B2	Genitive after indefinite article	1.00	0.60
3257	B2	Genitive after zero article	1.00	0.60
2282	A2	Perfect Tense	1.00	0.60

Table 3: Precision and recall scores for features below the 0.8 threshold in at least one metric.

4.2 Feature Validation

We evaluated the predictive performance of the extracted criterial features for regression as well as classification models, using EBM for both approaches. Results are reported in Table 4. In addition, we report global and feature-specific results from the EBM. In the regression setting, the model achieved a Root Mean Squared Error (RMSE) of 0.501 and a Mean Absolute Error (MAE) of 0.396. To enable comparison with the classification setup, regression predictions were rounded to the nearest CEFR level. Under this discretized evaluation, the model achieved an accuracy of 0.704, a macro-averaged F1 score of 0.667, a macro-averaged precision of 0.692, and a macro-averaged recall of 0.656. In the classification setting using an EBM, the model achieved an overall accuracy of 0.769. Macro-averaged F1 was 0.740, with a macro-averaged precision of 0.750 and a macro-averaged recall of 0.733. Direct comparison of the regression results and the classification model shows higher performance for the classification approach across all reported metrics. Subsequent analyses are therefore based on the classification model.

Category	Metric	Value
Regression	RMSE	0.501
	MAE	0.392
Classification	Accuracy	0.769
	F1 (macro)	0.740
	Precision (macro)	0.750
	Recall (macro)	0.733
Regression	Accuracy	0.711
	F1 (macro)	0.678
	Precision (macro)	0.724
	Recall (macro)	0.665

Table 4: Model performance metrics for regression, classification EBM, and rounded regression outputs.

Figure 1 shows the confusion matrix for the classification model. Consistent with previous findings (e.g., Lagutina et al., 2023), most errors occur between adjacent levels, especially at the intermediate level. This is especially interesting considering the class imbalance: As most data exists for the intermediate level, it would be expected that the model tends to overestimate this level, which is not the case for this model. Misclassifications spanning two levels are comparatively rare. Interestingly, recall for A1 level is surprisingly high, and the

confusion matrix shows that, if misclassified, intermediate texts are more frequently classified as easy texts, meaning that the models has a slight tendency to underestimate CEFR levels.

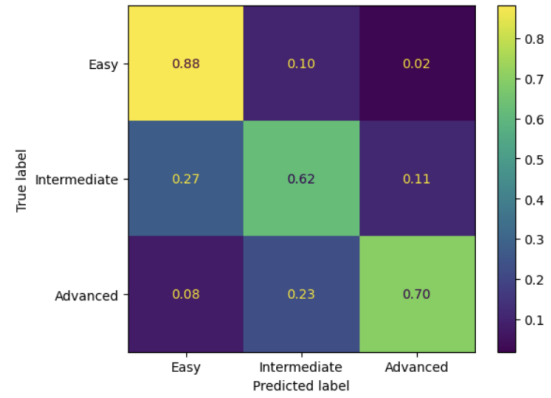


Figure 1: Confusion Matrix showing results in percent for the classification model.

It is important to note that there currently is no work on using German CFs as features for machine learning classifiers, however there is work using linguistic complexity measures. Our results are lower than classification results reported in previous work on linguistic complexity for German on the same corpus (Weiss et al., 2021), but are comparable to studies using criterial features for English classification tasks (Gaillat et al., 2022).

Figure 2 shows the development of the ten most strongly changing B1/B2 features across the three readability levels. Overall, most features show a

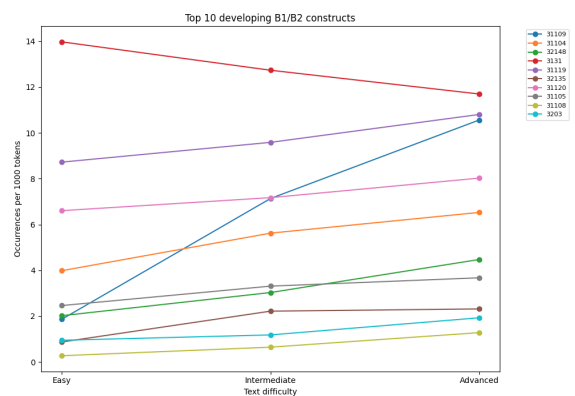


Figure 2: Developmental trajectories of B1 and B2 features with strongest development.

clear increase from easy to advanced texts, indicating that higher-level grammatical constructions tend to occur more frequently in more difficult texts. The overall trend suggests that the frequency of B1/B2 features increases with text difficulty, as

would be expected. The strongest increase is observed for feature 31109 (simple past), which rises substantially from easy to advanced texts. One interesting observation is the development of feature 3131 (comparative of B1-level adverbs), which shows a slight decrease across levels.

As can be seen in Figure 3, global term importances derived from the EBM (mean absolute weighted scores) indicate that features 31109 (simple past) and 2182 (perfect tense of A1 verbs) were the most influential predictors. These were fol-

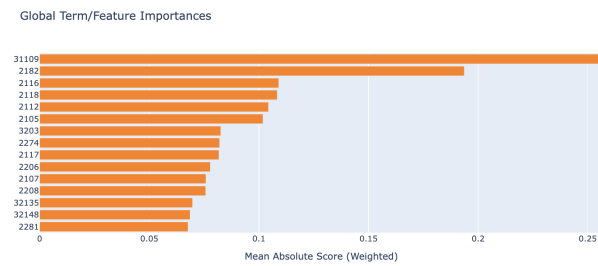


Figure 3: Global term importances for ten most informative features derived from the EBM.

lowed by features 2116 (A1 level verbs in present tense), 2118 (*to be* in present tense), 2112 (nominative of personal pronouns), 2105 (nominative of A1 nouns), 3203 (B2 level conjunctive adverbs), 2274 ("that" clauses), 2117 (*to have* in present tense), 2206 (relative pronouns in nominative case), 32135 (*to become* in passive present tense), 32148 (sub-junctive II), and 2281 (past participle).

Figure 4 shows the learned score contributions of feature 2182 and 31109. For feature 2182 (perfect tense of A1 verbs), the contributions vary across its observed frequency range (0–42). The distribution of this feature shows a concentration of observations in the lower frequency intervals. As the frequency of feature 2182 increases, the model increasingly favors class 1 (easy) while simultaneously decreasing the likelihood of class 3 (advanced), with class 2 (intermediate) exhibiting comparatively small changes. Thus, higher frequencies of feature 2182 are strongly associated with class 1 predictions, whereas low frequencies are more indicative of class 3.

For feature 31109 (simple past), score contributions vary across its observed frequency range (0–155), with most observations concentrated in the lower frequency intervals. In contrast to feature 2182, increasing frequencies of feature 31109 strongly increase the likelihood of class 3 (advanced) while reducing the likelihood of class 1



Figure 4: EBM shape functions and value distributions for the two most informative features. Top: Feature 31109. Bottom: Feature 2182. The upper panels show the learned score contributions across feature values, and the lower panels show the empirical value distributions.

(easy), again with comparatively smaller effects for class 2 (intermediate). High frequencies of feature 31109 therefore act as a strong indicator for class 3. As seen before, of all B1/B2 level features, feature 31109 also showed the strongest increase from easy to advanced texts. The results will be discussed in the following.

5 Discussion

This study addressed two research questions concerning the operationalization and predictive power of CFs for German. Overall, the analyses provide promising results that CFs can be operationalized computationally and that they capture meaningful aspects of language development.

Regarding RQ1, the results indicate that CFs can be reliably operationalized and automatically detected. The results across CEFR levels indicate that the system maintains very high precision across all proficiency levels, meaning that identified structures are highly reliable. The decrease in recall at higher levels likely reflects increasing linguistic complexity, which introduces greater structural variation and thus poses additional challenges. Moreover, 90% of individual features achieve precision and recall above 0.8. This demonstrates that the linguistic constructs defined in the GGP can be translated into computational rules and identified in language data. In addition, the use of a targeted test suite proved to be a useful methodology for

evaluating the performance of the extraction system, as it allowed us to systematically assess the performance of individual features and to improve the system cyclically, incorporating challenging negative examples.

With respect to RQ2, the results show that automatically detected CFs provide meaningful information for predicting CEFR levels. Several CFs were found to be highly informative for the EBM classifier. Notably, the most informative feature was the use of simple past (31109), followed by the perfect tense (2182), which involves the characteristic German split verb construction. Two additional features related to the split verb construction (2281, past participle and 32135, *to become* (*bekommen*) in passive present tense), also appeared among the most informative predictors. These findings suggest that verb-related features play an important role in distinguishing CEFR levels and reflect developmental patterns in second language acquisition.

An interesting observation concerns the distribution of the perfect tense feature of A1 level verbs (2182). While constructions involving the perfect tense and participle formation are theoretically related to the German split verb pattern, the model indicates that feature 2182 is particularly associated with easier texts, corresponding roughly to A2 levels. This appears to diverge from predictions made by PT, where verb-splitting phenomena are expected to emerge only at stage three (out of five stages) of the developmental hierarchy (Pienemann, 1998). One possible explanation is that certain perfect tense forms, particularly with high-frequency verbs such as *to be* (*sein*) and *to go* (*gehen*) (e.g., *I went*, *Ich bin gegangen*), may be introduced early in language instruction as formulaic constructions for communicative purposes. Learners may therefore acquire these forms initially as fixed expressions rather than as fully productive grammatical structures. In this sense, the presence of perfect tense forms in lower-level texts does not necessarily indicate that learners have acquired the underlying verb-splitting mechanism. Instead, the productive use of separable verb constructions may still develop later, in line with the predictions of developmental theories such as PT.

The results also show the importance of case marking (features 2112, 2105 and 2208, respectively, personal pronouns, nouns, and relative pronouns in nominative case), which showed as highly informative features in the classification model. In

PT, case marking is treated as a diacritic feature associated with a lexical item, rather than being a central developmental structure. However, previous research has highlighted the importance of morphological development for German L2 learners (Weiss, 2024), and the functional grammar underlying *Profile Deutsch* (Glaboniat, 2010) also assigns a prominent role to case marking. Our results provide further empirical evidence supporting this, suggesting that case marking contributes to distinguishing readability levels in graded readers.

As briefly mentioned in Section 4, classifiers based on linguistic complexity measures achieve higher performance on the same corpus than the approach presented in this paper (Weiss et al., 2021). This difference can be attributed to several factors. First, Weiss et al. (2021) extract a large set of over 300 features capturing multiple linguistic dimensions, including syntactic, lexical, and discourse-level properties. Second, linguistic complexity measures are inherently aggregative, which may be better suited for classification tasks. Finally, differences in model architecture, such as the use of Support Vector Machines (SVMs) compared to EBMs, may also contribute to performance differences. Nevertheless, criterial features offer distinct advantages that make them particularly valuable in educational contexts. In contrast to complexity measures, CFs provide a transparent and linguistically grounded representation of language, as they can be directly linked to CEFR levels and interpreted in terms of developmental trajectories. This allows for a more fine-grained and interpretable analysis of learner-relevant structures, which is not straightforward with aggregative complexity measures.

From a pedagogical perspective, the GGP and the PALME system enable targeted readability assessment by identifying whether a text is suitable for a specific learner group (e.g., B1 learners of German), while also highlighting specific linguistic structures that contribute to its difficulty. This opens up possibilities not only for text selection but also for diagnostic applications, such as identifying particularly challenging constructions within a text. At the same time, the two approaches are not mutually exclusive. Combining criterial features with linguistic complexity measures may lead to more robust and accurate classification systems, while preserving interpretability and pedagogical relevance.

6 Future Work

Several directions for future research emerge from the present study. First, the test suite developed for evaluating the automatic detection of CFs could be used to assess the capabilities of LLMs in identifying and analyzing linguistically motivated grammatical constructs. In particular, LLMs could be explored as a complementary approach in cases where rule-based detection proves difficult or insufficient. Another approach to address limitations of upstream tagging could be to explore the use of constituency parsing. Second, the test suite could be expanded to include authentic learner language data. Incorporating learner corpora would allow us to assess how well the system performs on naturally occurring learner language and to examine how CFs manifest in real learner production. Third, future work could investigate the relationship between automatically detected CFs and CEFR level assignments in more detail. Following approaches such as Verratti-Souto et al. (2025), statistical analyses would contribute to a better understanding of how closely criterial feature mappings correspond to actual proficiency development.

7 Conclusion

In this paper, we presented a systematic, test suite-based method to evaluate rule-based feature extractions systems by analyzing the reliability of the PALME system. We showed that PALME reliably extracts 90% of the extracted features with precision and recall above .8 using positive and negative examples. We moreover used the system to analyze graded reading material, thus contributing to interpretable ARA.

The findings indicate that criterial features capture aspects of developmental trajectories, but they also extend beyond them. PT predicts that certain grammatical phenomena emerge at specific stages of development, such as the split verb pattern associated with separable verb constructions. Our results partially align with these predictions. At the same time, the model also relies on a broader set of linguistic constructs, including morphological features. This suggests that criterial features not only reflect developmental stages but also capture additional dimensions.

8 Limitations

While the present study provides promising results, several limitations point to important directions for

future research. First, the evaluation of the PALME system is based on a manually constructed, corpus-informed test suite rather than a fully authentic corpus. Although this allows for controlled testing of individual features, it may not fully capture the variability of real-world language use. In addition, each feature is represented by only a small number of examples, which constrains the range of linguistic realizations covered.

Second, the system depends on upstream preprocessing components such as part-of-speech tagging and parsing. Errors in these components may propagate and affect feature detection, particularly for more complex constructions.

Third, the validation of criterial features is limited to a single corpus of graded readers, raising questions about the generalizability of the findings across different corpora and text types. Future work should therefore investigate cross-corpus performance to assess the robustness of the approach.

Finally, the study does not include learner language data. While graded readers approximate proficiency levels, they do not fully reflect actual learner production. Evaluating the system on learner corpora would therefore provide a more direct validation of criterial features as indicators of language development.

Acknowledgments

We would like to thank the LEAD Graduate School and Research Network for its support. Our research was supported by the WoLKE project (Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg) and the EU Project LATILL (Level-Adequate Texts in Language Learning, Reference number 2021-1-AT01-KA220-SCH-000029604). We also thank Sarina Doster for her contribution to the development of the test suite.

References

- Soroosh Akef, Detmar Meurers, Amália Mendes, and Patrick Rebuschat. 2025. [Interpretable Machine Learning for Societal Language Identification: Modeling English and German Influences on Portuguese Heritage Language](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 50–62.
- Lorna Balkan, Siety Meijer, Doug Arnold, Eva Dauphin, Dominique Estival, Kirsten Falkedal, Sabine Lehmann, Klaus Netter, and Sylvie Regnier-Prost. 1994. Issues in Test Suite Design. *Report to LRE*, 62:089.

- Kevyn Collins-Thompson. 2014. [Computational Assessment of Text Readability: A Survey of Current and Future Research](#). *ITL-International Journal of Applied Linguistics*, 165(2):97–135.
- Council of Europe. 2020. [Common European Framework of Reference for Languages: Learning, Teaching, Assessment – Companion Volume](#). Council of Europe Publishing, Strasbourg, France.
- Rocio Cuberos Vicente and Barbara De Cock. 2023. [Towards a Graded Lexical Inventory in L2 Spanish: Insights from Productive Vocabulary Knowledge](#). In *Proceedings of the Workshop on Linking Lexicographic and Language Learning Resources (4LR)*.
- Mark Davies. 2009. [The 385+ Million Word Corpus of Contemporary American English \(1990–2008+\): Design, Architecture, and Linguistic Insights](#). *International Journal of Corpus Linguistics*, 14(2):159–190.
- Thomas Gaillat, Andrew Simpkin, Nicolas Ballier, Bernardo Stearns, Annanda Sousa, Manon Bouyé, and Manel Zarrouk. 2022. [Predicting CEFR levels in learners of English: The Use of Microsystem Criterial Features in a Machine Learning Approach](#). *ReCALL*, 34(2):130–146.
- Manuela Glaboniat, editor. 2010. *Profile Deutsch: Gemeinsamer europäischer Referenzrahmen; Lernzielbestimmungen; Kannbeschreibungen; Kommunikative Mittel; Niveau A1-A2, B1-B2; C1-C2; [CD-ROM Version 2.0 mit Begleitbuch]*, nachdr. edition. Langenscheidt.
- John A. Hawkins and Paula Buttery. 2010. [Criterial Features in Learner Corpora: Theory and Illustrations](#). *English Profile Journal*, 1:e5.
- John A. Hawkins and Luna Filipović. 2012. [Criterial Features in L2 English: Specifying the Reference Levels of the Common European Framework](#), volume 1 of *English Profile Studies*. Cambridge University Press.
- Stephen D Krashen. 1985. *The input hypothesis: Issues and implications*. Longman, New York.
- Nadezhda Stanislavovna Lagutina, Kseniya Vladimirovna Lagutina, Anastasya Mikhailovna Brederman, and Natalia Nikolaevna Kasatkina. 2023. [Text Classification by CEFR Levels using Machine Learning Methods and BERT Language Model](#). *Modeling and Analysis of Information Systems*, 30(3):202–213.
- Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, and Eva Dauphin. 1996. [TSNLP - Test Suites for Natural Language Processing](#). In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 711–716.
- Denise Löfflad, Benedikt Beuttler, and Detmar Meurers. 2025. [German Grammar Profile for Learners: Pedagogical Feature Definition and Automated Extraction](#). In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, pages 212–223.
- Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. [Accurate Intelligible Models with Pairwise Interactions](#). In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*, pages 623–631.
- Salwa Mohamed. 2023. [The Development of an Arabic Curriculum Framework based on a Compilation of Salient Features from CEFR Level Descriptors](#). *The Language Learning Journal*, 51(1):33–47.
- Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. [InterpretML: A Unified Framework for Machine Learning Interpretability](#). *arXiv preprint arXiv:1909.09223*.
- Anne O’Keeffe and Geraldine Mark. 2017. [The English Grammar Profile of Learner Competence: Methodology and Key Findings](#). *International Journal of Corpus Linguistics*, 22(4):457–489. Publisher: John Benjamins Publishing Company Amsterdam/Philadelphia.
- Manfred Pienemann. 1989. [Is language teachable? psycholinguistic experiments and hypotheses](#). *Applied Linguistics*, 10(1):52–79.
- Manfred Pienemann. 1998. [Developmental Dynamics in L1 and L2 Acquisition: Processability Theory and Generative Entrenchment](#). *Bilingualism: Language and Cognition*, 1(1):1–20.
- Manfred Pienemann. 2015. [An outline of processability theory and its relationship to other approaches to SLA](#). *Language Learning*, 65(1):123–151.
- Rashmi Prasad and Anoop Sarkar. 2000. [Comparing Test-Suite based Evaluation and Corpus-based Evaluation of a wide-coverage Grammar for English](#). In *Proceedings of the Workshop on Using Evaluation within Human Language Technology Programs: Results and Trends. (LREC)*, pages 7–12, Athens, Greece.
- Nelly Sagirov and Xiaobin Chen. 2025. [POLKE: A System for Comprehensively Annotating Pedagogically-Oriented Grammatical Structure Use in Language Production](#).
- Angeliki Salamoura and Nick Saville. 2010. [Exemplifying the CEFR: Criterial Features of Written Learner English from the English Profile Programme](#). In Inge Bartning, Maisa Martin, and Ineke Vedder, editors, *Communicative proficiency and linguistic development: Intersections between SLA and language testing research*, volume 1 of *Eurosla Monographs Series*, pages 101–131. European Second Language Association.

- Andrea Vázquez-Ingelmo, Denise Löfflad, Alicia García-Holgado, Roberto Therón-Sánchez, and Detmar Meurers. 2026. [From CEFR Classification to Generative AI Materials: Designing and Validating the LATILL Platform](#). *Universal Access in the Information Society*, 25(2):67.
- Daniela Verratti-Souto, Nelly Sagirov, and Xiaobin Chen. 2025. [NLP-Powered Quantitative Verification of the English Grammar Profile’s Structure-Level Assignment](#). *Annual Review of Applied Linguistics*, pages 1–22.
- Zarah Weiss. 2024. [An Integrative Approach to Linguistic Complexity Analysis for German](#). PhD Thesis, Universität Tübingen.
- Zarah Weiss, Xiaobin Chen, and Detmar Meurers. 2021. [Using Broad Linguistic Complexity Modeling for Cross-Lingual Readability Assessment](#). In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 38–54. LiU Electronic Press.
- Zarah Weiss and Detmar Meurers. 2021. [Analyzing the Linguistic Complexity of German Learner Language in a Reading Comprehension Task: Using Proficiency Classification to Investigate Short Answer Data, Cross-Data Generalizability, and the Impact of Linguistic Analysis Quality](#). *International Journal of Learner Corpus Research*, 7(1):84–131.
- Katrin Wisniewski. 2020. [SLA Developmental Stages in the CEFR-Related Learner Corpus MERLIN: Inversion and Verb-End Structures in German A2 and B1 Learner Texts](#). *International Journal of Learner Corpus Research*, 6(1):1–37.

A Appendix

ID	Level	Feature	Example	Translation
2105	A1	Nominative of A1 Nouns	Die Frau arbeitet.	The woman is working.
2111	A1	Negation article 'kein' in the accusatives	Er sieht keinen Hut.	He has no hat.
2112	A1	Nominative of Personal Pronouns	Ich wohne in Tübingen.	I live in Tübingen.
2113	A1	Accusative of Personal Pronouns	Sie besucht ihn .	She visits him .
2116	A1	Present Tense of A1 Verbs	Wir fahren nach Hause.	We drive back home.
2117	A1	<i>to have</i> in Present Tense	Wir haben das nicht.	We do not have that.
2118	A1	<i>to be</i> in Present Tense	Es ist Donnerstag.	It is thursday.
2182	A1	Perfect Tense of A1 Verbs	Ich habe das Buch gelesen .	I have read the book.
2206	A2	<i>Nominative of Relative Pronouns</i>	Die Frau, die ich sehe.	The woman that I see.
2263	A2	Nominative of Adjectives after Zero Article	Kleine Kinder spielen dort.	Small children play there.
2264	A2	Accusative of Adjectives after Zero Article	Sie sollten frisches Obst essen.	They should eat fresh fruit.
2265	A2	Dative of Adjectives after Zero Article	Guten Freunden gibt man das.	You give that to good friends.
2268	A2	A2 Level Indefinite Pronouns	Das kann nicht jeder .	Not everyone can do that.
2274	A2	"that" Clauses	Ich hoffe, dass sie kommt.	I hope that she comes.
2281	A2	Past Participle	Wir haben dich besucht .	We visited you.
3131	B1	Comparative of B1 Level Adverbs	Er geht öfter als ich.	he goes there more often than me.
3132	B1	Superlative of B1 Level Adverbs	Er geht am öftesten dorthin.	He goes there more often .
3168	B1	B1 Level Indefinite Pronouns	Es gibt nur noch wenige .	There is only few left.
31109	B1	Simple Past	Sie wandte sich ab.	He turned away.
31120	B1	Genitive of Articles	Das Nest der Wespen.	The nest of the wasps. - The wasps' nest.
3134	B1	Adverbial Temporal Determinations	Er hat mit 21 geheiratet.	He got married at 21.
3203	B2	B2 Level Conjunctive Adverbs	Anschließend gingen wir weg.	We left afterwards .
3256	B2	Genitive after Indefinite Article	Das Nest eines jungen Spechts.	The nest of a young woodpecker.
3257	B2	Genitive after Zero Article	Der Geruch frischen Kaffees.	The smell of fresh coffee.
32135	B2	<i>to become</i> in Passive Present Tense	Der Brief wird geschrieben .	The letter is being written .
32148	B2	Subjunctive II for unrealistic Wishes	Wäre ich ein Vogel, flöge ich zu dir.	If I were a bird, I would fly to you.

Table 5: Feature descriptions from the GGP including an example sentence in German and translated to English.

B Appendix

ID	Pos1	Pos2	Pos3	Pos4	Pos5	Neg1	Neg2	Neg3	Neg4	Neg5	Precision	Recall
2168	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2165	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2166	1	1	1	1	0	0	1	0	0	0	0.8	0.8
2103	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2108	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2111	1	0	1	0	1	0	0	0	0	0	1.0	0.6
2110	1	1	1	0	1	0	0	0	0	0	1.0	0.8
2129	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2130	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2117	1	1	1	1	1	0	0	1	0	0	0.83	1.0
2118	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2119	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2120	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2121	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2122	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2123	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2124	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2125	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2126	1	1	1	0	1	0	0	0	0	0	1.0	0.8
2127	1	1	1	1	1	1	0	0	0	0	0.83	1.0
2128	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2101	1	1	1	1	1	1	1	0	0	0	0.71	1.0
2102	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2131	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2132	1	1	1	1	1	1	0	0	0	0	0.83	1.0
2161	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2162	1	1	1	1	1	0	0	1	0	0	0.83	1.0
2163	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2164	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2129	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2130	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2116	1	1	1	0	1	0	0	0	0	0	1.0	0.8
2146	1	0	1	1	1	0	0	0	0	0	1.0	0.8
2104	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2182	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2107	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2105	1	1	1	1	1	0	1	0	0	0	0.83	1.0
2106	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2133	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2112	1	1	1	1	1	0	0	0	0	1	0.83	1.0
2113	1	1	0	1	0	0	0	0	0	1	0.75	0.6
2114	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2115	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2109	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2169	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2134	1	1	1	0	1	0	0	0	0	0	1.0	0.8
2135	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2200	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2201	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2231	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2254	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2255	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2256	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2257	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2258	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2259	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2260	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2263	1	1	0	0	1	0	0	0	0	0	1.0	0.6
2264	1	0	1	0	0	0	0	0	0	0	1.0	0.4
2265	1	1	1	0	0	0	0	0	0	0	1.0	0.6
2261	1	1	0	1	1	0	1	0	0	0	0.8	0.8
2262	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2267	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2268	0	1	1	0	0	0	0	0	0	0	1.0	0.4
2206	1	1	1	1	1	0	0	0	0	0	1.0	1.0

ID	Pos1	Pos2	Pos3	Pos4	Pos5	Neg1	Neg2	Neg3	Neg4	Neg5	Precision	Recall
2207	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2274	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2275	1	1	1	1	1	0	0	0	1	0	0.83	1.0
2276	1	1	0	1	1	0	0	0	0	0	1.0	0.8
2277	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2279	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2280	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2281	0	1	0	1	1	0	0	0	0	0	1.0	0.6
2282	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2283	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2284	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2285	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2205	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2202	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2203	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2294	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2234	1	0	1	1	1	0	1	0	0	0	0.8	0.8
2235	1	1	1	1	1	0	0	0	0	1	0.83	1.0
2278	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2269	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2271	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2270	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2266	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2210	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2208	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2209	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2288	1	1	1	1	1	0	0	0	0	0	1.0	1.0
2286	1	1	1	1	0	0	0	0	0	0	1.0	0.8
2236	1	1	0	1	1	0	0	0	0	1	0.8	0.8
2287	1	1	1	0	1	0	0	0	0	0	1.0	0.8
2295	1	1	1	1	1	0	0	0	0	0	1.0	1.0
31103	1	0	1	1	1	0	0	0	0	0	1.0	0.8
31104	1	1	1	1	1	0	0	0	0	0	1.0	1.0
31105	1	1	0	1	1	0	0	0	0	0	1.0	0.8
3178	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3179	1	1	1	1	1	0	0	0	0	0	1.0	1.0
31108	1	1	1	1	1	0	0	0	0	0	1.0	1.0
31109	1	1	1	1	1	0	0	0	0	0	1.0	1.0
31110	0	1	1	1	1	0	0	0	0	0	1.0	0.8
31111	1	1	1	1	1	0	0	0	0	0	1.0	1.0
31112	1	1	0	1	1	0	0	0	0	0	1.0	0.8
31113	1	1	0	1	1	0	0	0	0	0	1.0	0.8
31115	1	1	0	1	1	0	0	0	0	0	1.0	0.8
31116	1	1	1	1	0	0	0	0	0	0	1.0	0.8
31114	1	1	1	1	1	0	1	0	0	0	0.83	1.0
3167	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3168	1	1	0	0	1	0	0	0	0	0	1.0	0.6
3104	1	1	1	0	1	0	0	0	0	0	1.0	0.8
31119	1	1	1	1	0	0	0	0	0	0	1.0	0.8
31120	0	1	1	1	0	0	0	0	0	0	1.0	0.6
3101	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3102	1	1	0	1	1	0	0	0	0	0	1.0	0.8
3103	1	0	1	1	1	0	0	0	0	0	1.0	0.8
3194	1	1	1	1	1	1	0	0	0	0	0.83	1.0
3134	1	0	1	1	0	0	0	0	0	0	1.0	0.6
3135	1	0	1	1	1	0	0	0	0	0	1.0	0.8
31130	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3105	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3106	1	1	1	1	1	0	1	0	0	0	0.83	1.0
3199	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3198	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3131	1	1	0	1	1	0	0	0	0	0	1.0	0.8
3132	0	0	0	1	0	0	0	0	0	0	1.0	0.2
3133	1	1	0	1	1	0	0	0	0	0	1.0	0.8
3166	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3136	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3201	1	1	1	1	1	0	0	0	0	0	1.0	1.0
32135	1	1	1	1	0	0	0	0	0	0	1.0	0.8

ID	Pos1	Pos2	Pos3	Pos4	Pos5	Neg1	Neg2	Neg3	Neg4	Neg5	Precision	Recall
32132	1	1	1	1	1	0	0	0	0	0	1.0	1.0
32133	1	1	1	1	1	0	0	0	0	0	1.0	1.0
32136	1	0	1	1	1	0	0	0	0	0	1.0	0.8
32137	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3255	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3256	0	1	1	0	1	0	0	0	0	0	1.0	0.6
3257	0	1	0	1	1	0	0	0	0	0	1.0	0.6
3267	0	1	1	1	1	0	0	0	0	0	1.0	0.8
3202	1	1	1	1	0	0	0	0	0	0	1.0	0.8
3203	1	1	1	1	1	0	0	0	0	0	1.0	1.0
32148	1	0	0	0	1	0	1	0	1	0	0.5	0.4
32149	1	1	1	1	0	0	0	0	0	0	1.0	0.8
32150	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3208	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3234	1	0	1	1	1	0	0	0	0	0	1.0	0.8
3235	0	1	1	1	1	1	0	0	0	0	0.8	0.8
3207	1	1	1	1	1	0	0	0	0	0	1.0	1.0
3231	1	1	0	1	1	0	0	0	0	0	1.0	0.8
3232	1	1	1	1	1	0	1	0	0	0	0.83	1.0
3236	0	1	1	1	1	0	0	0	0	0	1.0	0.8

Letting Tutor Personas Speak Up for LLMs: Learning Steering Vectors from Dialogue via Preference Optimization

Jaewook Lee¹, Alexander Scarlatos¹, Simon Woodhead², Andrew Lan¹

¹University of Massachusetts Amherst, ²Eedi

{ajscarlatos, jaewooklee, andrewlan}@cs.umass.edu, simon.woodhead@eedi.co.uk

Abstract

With the emergence of large language models (LLMs) as a powerful class of generative artificial intelligence (AI), their use in tutoring has become increasingly prominent. Prior works on LLM-based tutoring typically learn a single tutor policy and do not capture the diversity of tutoring styles. In real-world tutor–student interactions, pedagogical intent is realized through adaptive instructional strategies, with tutors varying the level of scaffolding, instructional directiveness, feedback, and affective support in response to learners’ needs. These differences can all impact dialogue dynamics and student engagement. In this paper, we explore how tutor personas embedded in human tutor–student dialogues can be used to guide LLM behavior without relying on explicitly prompted instructions. We train a steering vector using preference optimization: an activation-space direction that guides model responses toward specific tutor personas. We find that this steering vector captures tutor-specific variation across dialogue contexts, improving semantic alignment with ground-truth tutor utterances and increasing preference-based evaluations, while largely preserving lexical similarity. Analysis of the learned scaling coefficients further reveals interpretable structure across tutors, corresponding to consistent differences in tutoring behavior. These results demonstrate that activation steering offers an effective and interpretable way for controlling tutor-specific variation in LLMs using signals derived directly from human dialogue data.

1 Introduction

With the emergence of large language models (LLMs) as a powerful class of generative artificial intelligence (AI), their use in educational tutoring has become increasingly prominent, spanning both general-purpose systems (Google, 2023; OpenAI, 2023) and education-specific tutors (Carnegie Learning, 2024; Khan Academy, 2023). Recent

work has also connected LLMs with Knowledge Tracing (KT), using KT-derived or LLM-assisted student-state estimates for tasks such as student error simulation and mastery modeling in tutoring dialogues (Huang et al., 2026; Duan et al., 2026; Scarlatos et al., 2025a).

Approaches to building LLM tutors vary in how pedagogical behavior is specified and optimized. Many approaches simply prompt LLMs with dialogue policies and guardrails to elicit best-practice pedagogy, including reducing solution leakage, encouraging scaffolding, and enforcing Socratic-style questioning (Pal Chowdhury et al., 2024; Puech et al., 2025; Zhang et al., 2024). Other approaches instead train LLMs using supervised fine-tuning (SFT) to mimic human tutor behavior or use reinforcement learning (RL) on synthetic tutoring dialogues. These approaches optimize proxies of pedagogical quality, such as improving feedback correctness and alignment with tutoring goals (Scarlatos et al., 2024), directly optimizing for pedagogical alignment signals derived from preference data over tutoring behaviors (Sonkar et al., 2024), or leveraging pedagogical rubrics and LLM-based simulated students in-the-loop to encourage greater learning outcomes (Dinucu-Jianu et al., 2025; Scarlatos et al., 2025b).

Despite making improvements on pedagogical quality, these approaches only learn a single tutor policy and do not attempt to capture the diversity of tutoring styles. Figure 1 illustrates how real tutors differ in their instructional style when addressing the same problem. One noticeable difference is the use of emojis and affective language, which conveys emotional support and rapport. Another key difference is instructional approach: the first tutor provides increased emotional support and guides the student step-by-step, whereas the other provides minimal rapport-building and asks the student to test a solution, emphasizing student self-completion. Therefore, we ask a research question:

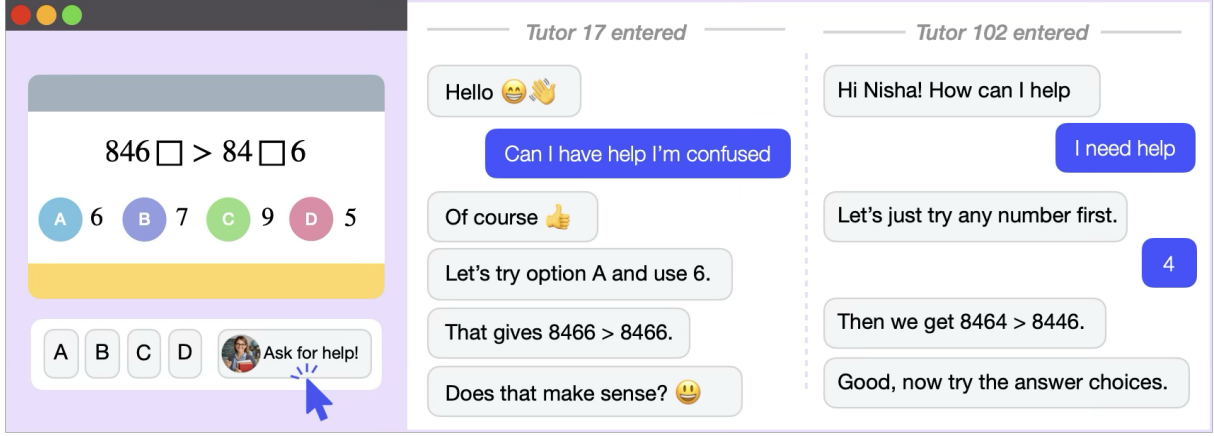


Figure 1: Real math tutoring dialogue on inequalities (Zent et al., 2025), showing how tutors 17 and 102 differ in instructional style when solving the same problem.

“how do we capture differences in tutoring styles and steer LLMs to exhibit these styles?”

1.1 Contributions

In this paper, we explore how implicit tutor personas embedded in human tutor-student dialogues can be used to guide LLM behavior, rather than relying on explicitly prompted instructions. We use preference optimization to learn a steering vector and a scaling coefficient that capture and control the persona direction in the model’s representation space. Using these learned parameters, we then steer the LLM in a tutor-specific manner. We find that our approach enables consistent recovery of tutor-specific steering across dialogue contexts, shifting the model away from population-mean behavior and toward the behavior exhibited by individual tutors. We further show the strength of this effect can be controlled at inference time through a global scaling factor, allowing for smooth adjustment of the intervention. Finally, analysis of the learned scaling coefficients reveals an interpretable structure over tutors that corresponds to variation in tutoring behavior, supporting activation steering as an effective way for capturing and controlling tutor-specific variation.

2 Related Work

2.1 Tutoring Strategies

Effective tutoring relies on a set of practical strategies rather than a single teaching approach, with tutors adjusting what they do in response to the learner’s needs. One core strategy is scaffolding, in which tutors provide support that helps learners solve problems they could not complete on their

own, and then gradually reduce that support as learners become more capable (Wood et al., 1976). Tutors also vary how directly they teach, moving between clear explanation or demonstration and guided practice that helps learners take on more responsibility over time (Rosenshine, 2012). In tutoring conversations, questioning strategies are used to prompt learners to explain their thinking while keeping the interaction focused on the learning goal (Graesser et al., 1995). During early stages of learning, tutors often use worked examples to show how problems can be solved and to make new skills easier to grasp (Sweller and Cooper, 1985). As learners practice, tutors provide formative, error-focused feedback to identify misunderstandings and support improvement (Shute, 2008). Finally, effective tutoring includes affective strategies such as encouragement, building rapport, and managing frustration in order to maintain engagement and persistence (D’Mello and Graesser, 2012).

2.2 Embedding Personas in LLMs

Research on embedding personas into LLMs generally follows two approaches: text-based conditioning and data-driven learning. In the text-based approach, personas are explicitly specified using structured descriptors such as the Five-Factor Model (Big Five) (McCrae and John, 1992), with prior work demonstrating that prompting an LLM with predefined Big Five trait profiles can induce consistent personality-aligned behaviors, which are validated by administering standardized personality questionnaires to the model (Serapio-García et al., 2023). In the context of intelligent tutoring systems, related work adopts explicit persona specifications to construct personality-aware student simulations

that combine cognitive attributes and personality traits, enabling systematic analysis of how different learner personas elicit and evaluate adaptive tutoring strategies (Liu et al., 2024b).

In contrast, data-driven learning induces personas from examples rather than explicit textual descriptions, and is commonly implemented either through SFT or by manipulating representation-level features (Chen et al., 2025; Mazaré et al., 2018). In this paper, we focus on activation steering using steering vectors, which estimate a direction from contrastive behaviors and inject that direction into a model’s hidden states to bias generation toward a target persona. A steering vector can be computed from differences in internal activations produced by paired inputs or outputs that reflect desired versus undesired traits, and then applied at selected layers and token positions during the forward pass to produce consistent behavioral shifts (Liu et al., 2024a; Rimsy et al., 2024; Turner et al., 2023; Wang and Shu, 2024). Our work follows BiPO, which itself builds on Direct Preference Optimization (DPO) by combining preference-based learning to train a steering vector with activation steering to achieve finer-grained control over model behavior (Cao et al., 2024; Rafailov et al., 2023). Like DPO, BiPO uses relative preferences between model outputs rather than absolute likelihood targets, yielding a contrastive learning signal that enables bidirectional steering between preferred and non-preferred behaviors.

3 Methodology

In this section, we detail our approach to learning a steering vector from tutor-student dialogues. We introduce the notation used throughout the paper:

$$\mathcal{D} \triangleq \bigcup_{i=1}^I \{(q_j, d_j^i)\}_{j=1}^{J_i},$$

$$d_j^i \triangleq (s_{j,0}, t_{j,1}^i, s_{j,1}, \dots, t_{j,K_j}^i, s_{j,K_j}).$$

$\mathcal{D}$ denotes a human tutor-student dialogue dataset, where I is the total number of tutors and $i \in \{1, \dots, I\}$ indexes a specific tutor. For each tutor i , J_i denotes the number of dialogues conducted by tutor i , and $j \in \{1, \dots, J_i\}$ indexes an individual dialogue. Each dialogue d_j^i is associated with a math question q_j . Each dialogue d_j^i consists of an alternating sequence of student turns $s_{j,k}$ and tutor turns $t_{j,k}^i$. $k \in \{1, \dots, K_j\}$ indexes (student, tutor) turn pairs in the dialogue, where K_j is the

total number of such turn pairs. The initial student turn $s_{j,0}$ is included when the student initiates the dialogue. We refer to the textual content of each turn as an *utterance*.

A steering vector encodes a direction in an LLM’s activation space, and when added to the activations induces changes in the LLM’s behavior. Learning a tutor-specific vector requires making explicit what the model is being steered away from and toward. In our setting, we define this direction relative to an *average* tutor: for a given dialogue context, the steering vector should shift the model from a population-mean tutor utterance toward the utterance produced by a particular tutor. We first SFT the LLM to condition on the dialogue context and generate an appropriate tutor response. The adaptation ensures that the model can perform the tutoring-response generation task before we estimate tutor-specific directions in activation space. We then use the fine-tuned model to generate a population-mean tutor utterance for each dialogue context and pair this generated utterance with the corresponding ground-truth tutor utterance. Finally, we learn a steering vector from these paired examples, so that the resulting vector captures tutor-specific stylistic variation beyond the population-average tutoring behavior. See Appendix Algorithm 1 for overall procedure to learn the steering vector.

3.1 Constructing Pairs

We use SFT to fine-tune an LLM to serve as a proxy for population-mean behavior, reducing response diversity and encouraging convergence toward a generic average-tutor behavior (Li et al., 2025); the resulting model is used to generate population-mean utterances. Let π_θ denote an LLM parameterized by θ ; we train π_θ via SFT by minimizing the negative log-likelihood of tutor utterances in $\mathcal{D}$ conditioned on the associated question q_j and the preceding dialogue context $(s_{j,\leq k}, t_{j,<k}^i)$:

$$\frac{1}{I} \sum_{i=1}^I \frac{1}{J_i} \sum_{j=1}^{J_i} \frac{1}{K_j} \sum_{k=1}^{K_j} -\log \pi_\theta(t_{j,k}^i | s_{j,\leq k}, t_{j,<k}^i, q_j).$$

For the preceding dialogue context, we construct a pair in which the ground-truth tutor utterance is the preferred response and a response generated by the SFT model is the dispreferred response. The SFT-generated response serves as a proxy for population-mean tutor behavior, so each pair explicitly defines the direction from the average tutor

response $\bar{t}_{j,k}$ toward the tutor-specific response $t_{j,k}^i$. See Appendix Tables 5 and 6 for examples of such contrastive pairs.

3.2 Learning a Steering Vector

We now detail what a steering vector is and how we learn it using the pairs. We introduce a steering vector v , shared across tutors, which we use to modify the model’s activation during inference by adding it to the latent activations $A_L(\cdot)$ at layer L . We assign each tutor a positive scaling coefficient δ_i that scales a shared steering direction v . We constrain these coefficients to have unit mean across tutors to eliminate a scale ambiguity, since the intervention depends only on the product $\delta_i v$ and could otherwise be left unchanged by scaling δ_i up while scaling v down (or vice versa) without affecting the loss. To parameterize δ_i while enforcing positivity, we introduce a tutor-specific learnable scalar u_i and define $\delta_i = \exp(u_i) / (\frac{1}{T} \sum_{m=1}^T \exp(u_m))$. We jointly optimize each u_i and the shared steering direction v in training. We initialize $u_i = 0$ for all tutors ($\delta_i = 1$), and initialize v from a standard Gaussian scaled by 0.5 to ensure a sufficiently strong initial steering magnitude. We note that while we learn the steering vector through preference optimization inspired by BiPO (Cao et al., 2024), we differ in that, rather than using bidirectional steering, we learn a scaling coefficient that scales a shared steering direction:

$$\min_{v,u} \mathbb{E}_{i,j,k} \left[-\log \sigma \left(\beta \log \frac{\pi_{\theta}(t_{j,k}^i | A_L(\cdot) + \delta_i v)}{\pi_{\theta}(t_{j,k}^i | A_L(\cdot))} \right. \right. \\ \left. \left. - \beta \log \frac{\pi_{\theta}(\bar{t}_{j,k} | A_L(\cdot) + \delta_i v)}{\pi_{\theta}(\bar{t}_{j,k} | A_L(\cdot))} \right) \right].$$

For tutor i , dialogue j , turn k , our objective includes two relative preference terms. The first log term encourages the steered LLM to have higher likelihood in generating actual tutor utterance $t_{j,k}^i$ compared to the unsteered SFT model. The right term does the exact opposite, encouraging the steered model to have lower likelihood in generating the population-average tutor utterance. All likelihoods in these terms are sequence log-likelihoods computed by summing next-token log-probabilities over the response tokens only, with prompt tokens masked out. The logistic loss encourages learning a shared steering direction v and tutor-specific coefficients δ_i such that applying $\delta_i v$ consistently favors tutor-specific outputs across tutors, dialogues, and turns. The hyperparameter β encourages similarity

to the base model’s behavior without the steering vector applied. We optimize this objective by iterating for T steps until the loss converges.

At test time, we introduce a global scaling factor α and apply activation steering as $A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v$. We observe that $\alpha = 1$ leads to longer utterances, because of the objective uses unnormalized sequence log-likelihoods, so the log-ratio accumulates over all response tokens. Consequently, longer tutor utterances can contribute larger preference margins than shorter ones, making the learned intervention at $\alpha = 1$ overly strong on continuing generation rather than stopping. While the training objective identifies a shared steering direction v and relative tutor-specific strengths δ_i with respect to the unsteered SFT model, a separate scaling factor is necessary to control the overall strength of the intervention during inference.

4 Experimental Settings

4.1 Dataset

We use the Question-Anchored-Tutoring-Dialogues-2k dataset (Zent et al., 2025), containing dialogues from web-based conversations between expert tutors and real students in an online math learning platform. The conversations are initiated by tutors or students requesting assistance while working through a lesson. We split the dataset into training, validation, and test sets at the dialogue level using an 80/10/10 split for each of the 21 unique tutors. On average, each tutor has 74.19 ($\sigma = 75.47$) dialogues in the training set, 8.90 ($\sigma = 9.45$) dialogues in the validation set, and 10.10 ($\sigma = 9.42$) dialogues in the test set. The average number of turns per dialogue is 11.75 ($\sigma = 1.10$) in training, 12.06 ($\sigma = 2.43$) in validation, and 12.25 ($\sigma = 1.73$) in test.

4.2 Evaluation Setup and Metrics

We evaluate how closely the steered tutor utterances match the corresponding ground-truth on a per-tutor basis, along two main axes: *lexical similarity* and *semantic similarity*. For each tutor i , we compare steered outputs $t_j^{\delta_i}$ to ground-truth utterances t_j across turns j . For lexical similarity, we measure surface-form overlap using ROUGE-L (R) (Lin, 2004) and BLEU(B) (Papineni et al., 2002). For semantic similarity, we assess semantic alignment using cosine similarity in embedding space (C) and a pairwise preference-based win rate

(W). We define $C(\delta_i) = \mathbb{E}_j [\cos(\phi(t_j^{\delta_i}), \phi(t_j))]$, where $\phi(\cdot)$ denotes a sentence embedding function from a pretrained SentenceBERT encoder (Reimers and Gurevych, 2019). For Win Rate, we use LLM-as-a-judge with Prometheus-Eval, given a dialogue and a ground-truth turn and chooses which of the steered or unsteered utterances more closely aligns with the tutoring style and language used (Kim et al., 2024). We report the win rate over unsteered SFT utterances, where 0.5 indicates the two perform equally. We do not report ties because they are rare. All metrics are computed per tutor and then averaged across tutors.

4.3 Model Parameters

We use Llama-3.1-8B-Instruct as the base model for SFT, which we adapt using LoRA (Hu et al., 2022) with a learning rate of 5×10^{-5} over 3 epochs, setting the LoRA rank to $r = 32$ and the scaling factor to $\alpha = 64$. Our SFT setup follows Scarlatos et al. (2026), adopting their prompts and training procedure. For learning v and u_i , we use the validation set, applying each u_i to the dialogues corresponding to tutor i and selecting hyperparameters by tracking validation loss; we set $\beta = 1.0$ and learning rate = 0.01, and apply activation steering at the final transformer layer (layer 32). Text generation uses a temperature of 1.0 and top- p sampling with $p = 0.95$, yielding a final validation loss of 0.543 at $T = 17$. The learned u_i have a mean of 0.053 and a standard deviation of 0.106. The activation steering mechanism follows the persona vector approach (Chen et al., 2025), where the learned steering vector is added to the layer activations at all token positions. All training and evaluation are conducted on a single NVIDIA A40 GPU with 48 GB of VRAM.

5 Experimental Results

5.1 Quantitative Results

We quantitatively evaluate activation steering for tutoring along two dimensions: (1) how well it recovers tutor-specific behavior at different stages of a dialogue, and (2) how this behavior varies with the strength of the steering intervention. Specifically, we first compare steered tutor utterances $\hat{t}_j$ with population-mean utterances $\bar{t}_j$, using ground-truth tutor utterances t_j as reference, in different stages of the dialogue (Table 1). We then analyze the effect of varying the global scaling factor α on both stylistic alignment and lexical similarity

Table 1: Comparison of population-mean utterances $\bar{t}_j$ and steered tutor utterances $\hat{t}_j$ against the ground-truth tutor utterances t_j across dialogue stages. Metrics are computed per tutor and then averaged across tutors.

Stage	Count	Method	R [†]	B [†]	C [†]	W [†]
early	326	$\bar{t}_j$	0.285	0.070	0.392	–
		$\hat{t}_j$	0.206	0.041	0.321	0.571
mid	1971	$\bar{t}_j$	0.165	0.019	0.385	–
		$\hat{t}_j$	0.157	0.018	0.426	0.587
late	326	$\bar{t}_j$	0.157	0.025	0.330	–
		$\hat{t}_j$	0.121	0.017	0.349	0.564

Table 2: Comparison of varying the strength of the steering vector by applying a scaling factor α in activation steering for all stages: $A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v$.

Count	Method	α	R [†]	B [†]	C [†]	W [†]
2623	$\bar{t}_j$	0.0	0.179	0.026	0.379	–
		0.3	0.181	0.027	0.400	0.536
	$\hat{t}_j$	0.5	0.187	0.028	0.407	0.539
		0.7	0.176	0.026	0.404	0.562
		1.0	0.159	0.021	0.403	0.582

(Table 2).

We conduct a stage-wise analysis because different parts of a dialogue serve different purposes. Early and late turns are largely greetings, encouragement, and closing remarks, which tend to follow common conventions and thus provide limited signal about a tutor’s instructional style. In contrast, the middle portion of the dialogue contains the core problem-solving interaction, where tutors explain concepts, provide hints, and respond to student errors. These turns are where tutor-specific strategies and language use are most evident. For this reason, we define the *mid stage* as the 10–90% interval of turns within each dialogue and focus our analysis on this region.

Table 1 shows the results on all metrics. Beyond early dialogue stages, steering leads to a clear improvement in semantic alignment with the ground-truth tutor: in the mid stage, cosine similarity increases from 0.385 for $\bar{t}_j$ to 0.426 for $\hat{t}_j$, and the LLM judge prefers $\hat{t}_j$ over the unsteered baseline in 58.7% of comparisons. At the same time, lexical overlap is only slightly reduced; ROUGE-L decreases from 0.165 to 0.157 and BLEU remains almost unchanged (0.019 to 0.018). This result suggests that steering improves tutor-specific style primarily by altering phrasing and discourse struc-

ture rather than by matching exact word sequences. Only in the early stage, the cosine similarity for $\hat{t}_j$ is worse than $\bar{t}_j$. The result is likely due to greetings in the early turns being inherently generic, which means that adding the steering vector over-exaggerates the effect of tutor style in these turns. A solution may be dynamically setting α to lower values for turns that are intended to be more generic.

Table 2 examines the effect of steering strength aggregated across all dialogue stages. Moderate values of α (0.3 and 0.5) slightly improve ROUGE-L and BLEU relative to the population-mean baseline, while also increasing cosine similarity. As α increases, Win Rate rises steadily from 0.536 to 0.582, indicating stronger expression of tutor-specific behavior. However, higher values of α also lead to lower ROUGE-L and BLEU scores, reflecting a trade-off in which stronger steering enhances stylistic alignment at the cost of lexical similarity.

Based on these results, we select $\alpha = 0.5$ for our qualitative analysis, since it provides a good balance between improving semantic and stylistic alignment with the target tutor while largely preserving lexical similarity.

5.2 Qualitative Results

5.2.1 Interpreting the steering direction

We visualize the scaling coefficient δ_i in Figure 2. To examine how δ_i correlates with tutoring styles or language use, we put tutors into three groups that each address a common math question in the dataset¹: from the left end to the middle (Tutors 283, 346, and 102), from the middle to the right (Tutors 283 and 1430), and from the middle to the right end (Tutors 19 and 2488). Table 3 provides an overview of the math questions and ground-truth utterances, where tutors with lower δ_i are placed on the left and those with higher δ_i are placed on the right.

Table 3 shows three example math questions (ratios, linear sequences, and order of operations) together with short tutor-student dialogue snippets. The table is organized to make the learned direction δ_i interpretable: tutors are arranged from left to right according to their position in the heatmap ordering (e.g., 283 $\rightarrow$ 346 $\rightarrow$ 102 $\rightarrow$ 1430 $\rightarrow$ 19 $\rightarrow$ 2488). For each question, the “Lower δ_i ” column contains tutors closer to the left side of this ordering, while the “Higher δ_i ” column contains tutors closer to the right side. The blue-highlighted

text marks the target tutor utterance to be predicted. Therefore, snippets show how the same dialogue context elicits different conversational moves and tutor utterances, as we move along the δ_i direction.

Tutor 283 who appears at the left end shows relational and pedagogical investment through explicit encouragement and step-by-step scaffolding. In the ratios example, Tutor 283 uses affective markers and praise (“:-”), “Yes, great!”) and then connects the student’s correct operation on the first term to the required operation on the second term: “the 3 in the ratio has been divided by 3, so you now need to divide the 2 by 3 as well.” In the linear sequence example, the tutor validates the student (“Great!”), explains the structural role of $4n$, and checks understanding with a concrete substitution prompt (“When $n = 1$, $4n = ?$ ”). These turns prioritize both student confidence and conceptual clarity.

Tutors positioned toward the middle of the δ_i direction, including Tutors 102, 1430, and 19, share a more efficiency-oriented style. Across examples, these tutors rely on short, task-focused prompts and corrective feedback, with reduced affective language and limited elaboration. Tutor 102 narrows the student’s response space through a fill-in-the-blank equation (“ $3 \div ___ = 1$ ”) and provides brief correction (“Almost, try again?”). Tutor 1430 advances the linear sequence problem through a series of concise checkpoints (“Great, now pattern 2?”, “Great, now pattern 3?”) and a minimal continuation of the sequence (“6, 10, 14,”). Tutor 19 similarly emphasizes efficient progression by providing succinct procedural guidance (“we multiply before we add”) and a short conceptual reminder (“Remember the fraction line means divide”), occasionally accompanied by a small positive signal with thumbs up emoji. Together, these tutors demonstrate consistent pedagogical engagement while prioritizing error diagnosis and progress over extended explanation or rapport-building.

At the right end of the δ_i direction, Tutor 2488 closely aligns with a low-investment style, focusing on just getting the task completed. The tutor gives short answers (“It means divide”) and follows up with a confusing justification (“it’s a weird one because you’d normally do division before the addition”), which does not clearly resolve the student’s uncertainty about expression structure. Compared to Tutor 19’s concise but accurate guidance, Tutor 2488’s response prioritizes moving the interaction forward, even at the expense of clarity in their explanation. Taken together, these examples

¹Question IDs: 104637, 91697, and 77628, respectively.

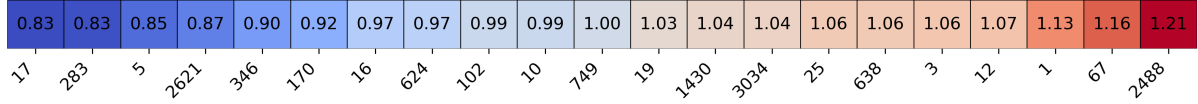


Figure 2: Heatmap of δ_i across 21 tutors with different tutoring styles.

Table 3: Examples of tutor-student dialogue across different math topics. Blue denotes our target utterance.

Question	Lower δ_i	Higher δ_i
Ratios <i>Lila and Elijah share some sweets in the ratio 3 : 2. What is this ratio in the form 1 : n?</i> A. 1 : 1.5 B. 1 : $\frac{2}{3}$ C. 1 : 0 D. 1 : 6	Tutor 283 T: :-) So, you need to find an equivalent ratio to 3 : 2. That is, instead 1 : n S: im not sure T: So, what would you divide 3 by, to get the answer 1? S: 3 divide by 3 T: Yes, great! So, the 3 in the ratio has been divided by 3, so you now need to divide the 2 by 3 as qwell. *well	Tutor 102 T: Ok let's take a look. What do we need to divide by to get the start of the ratio down from 3 to 1? S: i dont know T: $3 \div \underline{\quad} = 1$ S: 1 T: Almost, try again?
Linear sequence <i>How many dots would there be in Pattern n ? [Image: 6 dots, 10 dots, 14, and 18 dots]</i> A. $4n + 2$ B. $4n - 2$ C. $2n + 4$ D. $n + 2$	Tutor 283 T: Yes of course, let's have a look So, how many dots are being added each time? S: 4? T: Great! So that tells us that the first part of our formula will be $4n$ S: so does the $4n$ refer to how many dots are being added each time? T: Yes, that's how we work out that bit Then, we need to see if that works, or if we need to add/subtract anything When $n=1$, $4n = ?$	Tutor 1430 T: Great, now pattern 2? S: 10 T: Great, now pattern 3? S: 14 T: So what is the pattern going up in? S: 4 T: 6, 10, 14,
Order of operations <i>What is the value of $\frac{1+3 \times 4}{2}$?</i> A. 12.5 B. 8 C. 7 D. 6.5	Tutor 19 T: Yep no worries With a fraction like this we would need to calculate the value of the numerator first What is $1 + 3 \times 4$ S: well 1 add 3 is 4 and 4 times 4 is 16 so will it be $16 <$ T: Not quite. Remember with the order of operations we multiply before we add so we need to do 3×4 first then add 1 S: oh 3 times 4 is 12 add the 1 is 13 T: 👍 So now we have $13/2$ Remember the fraction line means divide	Tutor 2488 S: What do I do after 3×4 T: With the fraction line what does that actually mean? S: I'm not sure what it means T: It means divide S: So 12 divided by 2 T: So it's a weird one because you'd normally do division before the addition but because it's all of the $1+12$ divided by 2

indicate that the learned direction δ_i corresponds to changes in tutoring style, from rapport-rich scaffolding on the left, through diagnostic instruction in the middle, to procedural assistance that emphasizes task completion over teaching on the right.

5.2.2 Case studies

Table 4 shows predicted utterances from unsteered SFT and steered generations with $\alpha = 0.5$. Overall, steering makes the generations better aligned with the ground-truth tutor utterances in Table 3, particularly in how the LLM matches the tutoring approach associated with δ_i . In many cases, the unsteered model produces reasonable tutoring moves, but it often misses the specific dialogue act or tone that distinguishes lower- δ_i tutors (rapport and scaffolding) from higher- δ_i tutors (corrective or short/concise continuations). With $\alpha = 0.5$, the model

more consistently reproduces the intended style difference between the two ends, yielding utterances that are closer to the ground-truth utterances in both tutoring style and language use.

In the ratios example, the ground-truth utterance of Tutor 283 first affirms the student and then explains that whatever is done to the 3 must also be done to the 2. The unsteered prediction (“Amazing!! And 2 divide by 3?”) has a friendly tone, but moves directly to $2/3$ without introducing additional steps. The steered prediction, however, provides a conceptual step while maintaining a friendly tone, thereby exhibiting a tutoring style similar to the ground truth. The ground-truth utterance of Tutor 102 is a brief correction (“Almost, try again?”) following an incorrect student answer. The unsteered prediction instead shifts into an elaborated instructional explanation (“So we need to divide

Table 4: Predicted utterances for the target utterances shown in Table 3.

α	Lower δ_i	Higher δ_i
0.0	Amazing!! And 2 divide by 3?	So we need to divide both sides by 3 $3 \div 3 = 1$ What would the ratio become?
0.5	And what would happen to 2? 😊 $3 \div 3 = 1$ So, we would have to divide both parts by 3 $2 \div 3 = ?$	Nearly, not 1... What do we need to divide by to get from 3 to 1?
0.0	Exactly, so if it's pattern 1 it will have $4n$ dots in the pattern	Great, so to get pattern n , we have $4(n)$ How many is it missing?
0.5	$4n$ refers to the number of times 4 is being added So, $4n$ means the number of terms we're adding 4 to And then, we have to account for the fact that in the first pattern ($n=1$) there are 6 dots, not 4 So, we start with $4n$ and then add 2	So in pattern 1 how many were there and what is this times by??
0.0	So when we have $1 + 3 \times 4$, this is different from $3 \times (1 + 4)$ Are you ok with why that's different?	You have to carry out both operations
0.5	brilliant Now we have $13 / 2$	Yes 👍

both sides by 3 ... What would the ratio become?"), which does not match Tutor 102's concise corrective move. The steered prediction "Nearly, not 1... What do we need to divide by to get from 3 to 1?" is much closer in style: it is explicitly corrective ("Nearly, not 1...") and prompts the student to retry the same subproblem, matching the ground-truth tutor strategy even if the exact wording differs.

In the linear sequence example, there is a similar pattern. The ground-truth utterance of Tutor 283 answers the student's clarification question about what $4n$ represents and then immediately transitions into a structured check ("When $n = 1$, $4n = ?$ "). The unsteered prediction ("Exactly, so if it's pattern 1 it will have $4n$ dots ...") moves the problem forward but does not directly address the student's question, which differs from the tutor's explanatory move. The steered prediction shows an explicitly explanatory turn that attempts to clarify what $4n$ means and then addresses it by referencing the mismatch at $n = 1$, introducing an offset ("we start with $4n$ and then add 2"). While the steered text is more verbose and uses some slightly awkward phrasing, it is closer in tutoring strategy because it responds to the student's conceptual query and uses a concrete consistency check to motivate the adjustment, which is the same pedagogical move that Tutor 283 uses. The ground-truth utterance of Tutor 1430 is extremely concise ("6, 10, 14,"), functioning more like a checkpoint or continuation than an explanation. The unsteered prediction introduces additional scaffolding ("How many is it missing?"), and the steered prediction becomes even more diagnostic ("So in pattern 1 how many were there and what is this times by?"). Neither matches the concise tutoring style in the ground truth, but the steered utterance aligns better in that it remains concise and task-driven.

In the order of operations example, steering yields the clearest improvement. The ground-truth utterance of Tutor 19 ends with a short affirmation with a procedural reminder, "(thumbs up) So now we have $13/2$ Remember the fraction line means divide." The unsteered prediction instead introduces a conceptual aid, contrasting two different expressions ("this is different from $3 \times (1 + 4)$ "), which is pedagogically plausible but does not match the ground-truth target's concise confirmation-and-next-step structure. The steered prediction shows "brilliant Now we have $13 / 2$," which is substantially closer in both tone and function, echoing the short praise and the immediate transition to the next step. The ground-truth utterance of Tutor 2488 is a long, somewhat muddled explanation about why the fraction is "weird," a term rarely used to explain order of operations. Therefore, both predictions struggle, and the steered one incorrectly assesses the student response as correct. The unsteered prediction is overly generic while the steered shows an acknowledgment, saying yes.

Across all examples, steering consistently improves alignment between model-generated utterances and ground-truth ones, capturing the specific tutoring strategy, tone, and style along the δ_i spectrum that unsteered SFT often misses. With $\alpha = 0.5$, steered outputs more reliably reflect the intended tutoring stance and underlying tutoring function, such as rapport-building and scaffolding, affirming before explaining, or concise, corrective continuations, suggesting steering effectively controls pedagogical intent beyond surface style.

6 Conclusions and Future Work

In this paper, we explored how implicit signals in human tutor-student dialogues can be used to guide LLMs towards different tutoring styles. We proposed an activation steering method that learns a shared steering direction and tutor-specific coefficients from real tutor-student dialogues, relative to a population-mean reference, enabling personalized, controlled tutor-specific behavior. We show that the learned steering direction and strengths lead to improved replication of individual tutor behavior. We also qualitatively show that it encodes changing tutoring styles from scaffolded, rapport-building through efficient, task-focused guidance to minimal, task-completion-oriented instruction.

There are several directions for future work. First, although our objective captures tutor-specific behavior via preferences, it does not explicitly encode pedagogical principles. Integrating pedagogical rubrics, learning-theoretic goals, or student outcome-based rewards could better align models with effective tutoring practices. Second, beyond steering tutor behavior, future work could learn complementary steering vectors for student models, enabling simulated tutor-student interactions driven by latent student personas. This would allow analysis of how interaction dynamics and tutor-student compatibility influence learning outcomes. Finally, extending the method to learn multiple steering directions, for example through low-rank subspaces from dimensionality reduction, could enable more fine-grained and disentangled control over distinct pedagogical attributes.

Acknowledgments

This work is partially supported by Renaissance Philanthropy under the learning engineering virtual institute (LEVI) initiative and the NSF under grants 2153481 and 2237676.

Limitations

There are several practical limitations to our work. First, we evaluate our method on a single math dialogue dataset, so it is unclear how well the results generalize to other datasets, domains, or tutoring settings. Second, our baselines are limited to supervised fine-tuning methods; future work should compare against prompting-based approaches as well. Third, we do not include human evaluation, so our findings rely on automatic metrics and model-based judgments.

Ethical Considerations

Our experiments use the Eedi dataset (Zent et al., 2025), in which dialogues are associated with pseudonymous identifiers rather than real names. We therefore do not identify any individual tutors or students in this work. Beyond this, we do not foresee additional ethical concerns specific to our experimental setup.

References

- Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *Advances in Neural Information Processing Systems*, 37:49519–49551.
- Carnegie Learning. 2024. Livehint overview. Online: <https://support.carnegielearning.com/help-center/math/livehint/article/livehint-overview/>.
- Runjin Chen, Andy Ardit, Henry Sleight, Owain Evans, and Jack Lindsey. 2025. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*.
- David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 272–292.
- Zhangqi Duan, Nigel Fernandez, and Andrew Lan. 2026. Kaser: Knowledge-aligned student error simulator for open-ended coding tasks. *arXiv preprint arXiv:2601.06633*.
- Sidney D’Mello and Art Graesser. 2012. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157.
- Google. 2023. Gemini. <https://deepmind.google/technologies/gemini/>. Accessed: 2026-01-21.
- Arthur C Graesser, Natalie K Person, and Joseph P Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied cognitive psychology*, 9(6):495–522.
- 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*.
- Shuyan Huang, Alexander Scarlatos, Jaewook Lee, and Andrew Lan. 2026. Interpretable difficulty-aware knowledge tracing in tutor-student dialogues. *Preprint*, arXiv:2605.01097.

- Khan Academy. 2023. Supercharge your teaching experience with khanmigo. Online: <https://www.khanmigo.ai/>.
- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.
- Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024a. In-context vectors: making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Zhengyuan Liu, Stella Xin Yin, Geyu Lin, and Nancy Chen. 2024b. Personality-aware student simulation for conversational intelligent tutoring systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 626–642.
- Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. *arXiv preprint arXiv:1809.01984*.
- Robert R McCrae and Oliver P John. 1992. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215.
- OpenAI. 2023. Chatgpt. <https://chat.openai.com>. Accessed: 2026-01-21.
- Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. 2024. Autotutor meets large language models: A language model tutor with rich pedagogy and guardrails. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 5–15.
- 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.
- Romain Puech, Jakub Macina, Julia Chatain, Mrinmaya Sachan, and Manu Kapur. 2025. Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26291–26311.
- 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. *Advances in neural information processing systems*, 36:53728–53741.
- 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.
- Nina Rimskey, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.
- Barak Rosenshine. 2012. Principles of instruction: Research-based strategies that all teachers should know. *American educator*, 36(1):12.
- Alexander Scarlatos, Ryan S Baker, and Andrew Lan. 2025a. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th international learning analytics and knowledge conference*, pages 249–259.
- Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. 2026. Simulated students in tutoring dialogues: Substance or illusion? *arXiv preprint arXiv:2601.04025*.
- Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025b. Training llm-based tutors to improve student learning outcomes in dialogues. In *International Conference on Artificial Intelligence in Education*, pages 251–266. Springer.
- 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*, pages 280–294. Springer.
- Gregory Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. 2023. Personality traits in large language models.
- Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research*, 78(1):153–189.
- Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard Baraniuk. 2024. Pedagogical alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13641–13650.
- John Sweller and Graham A Cooper. 1985. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and instruction*, 2(1):59–89.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Haoran Wang and Kai Shu. 2024. Trojan activation attack: Red-teaming large language models using steering vectors for safety-alignment. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, page 2347–2357. Association for Computing Machinery.

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.

Matthew Zent, Digory Smith, and Simon Woodhead. 2025. PIIvot: A lightweight NLP anonymization framework for question-anchored tutoring dialogues. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27467–27476.

Liang Zhang, Jionghao Lin, Ziyi Kuang, Sheng Xu, and Xiangen Hu. 2024. Spl: a socratic playground for learning powered by large language model. *arXiv preprint arXiv:2406.13919*.

A Appendix

As mentioned earlier, our prompting and supervised fine-tuning (SFT) approach builds on the framework of [Scarlatos et al. \(2026\)](#), using the code available on <https://github.com/umass-ml4ed/sim-student-eval>. In this section, we outline the setup required to train a steering vector.

A.1 Implementation Details

We implement the method in two stages: (i) construction of turn-level preference pairs from tutor-student dialogues, and (ii) optimization of a shared steering direction v together with tutor-specific scaling coefficients $\{\delta_i\}_{i=1}^I$. The implementation follows the formulation in the main text, but includes several practical choices for efficient training and evaluation.

Sequence log-likelihoods. Given a prompt-response pair (q, r) , we compute the conditional sequence log-likelihood

$$\log \pi_\theta(r \mid q)$$

by summing next-token log-probabilities over the response tokens only. Prompt tokens are masked out and do not contribute to the score. These log-likelihoods are computed both for the unsteered

model and for the steered model. Since the baseline values

$$\log \pi_\theta(t_{j,k}^i \mid q_{j,k}^i), \quad \log \pi_\theta(\bar{t}_{j,k} \mid q_{j,k}^i)$$

are reused throughout optimization, they are pre-computed and cached for efficiency.

Parameterization of tutor-specific strength. As in the main text, we learn a shared steering direction v and tutor-specific coefficients δ_i . To enforce positivity and remove scale ambiguity, we optimize unconstrained parameters $u_i \in \mathbb{R}$ and set

$$\delta_i = \frac{\exp(u_i)}{\frac{1}{I} \sum_{m=1}^I \exp(u_m)}.$$

This ensures $\delta_i > 0$ and $\frac{1}{I} \sum_{i=1}^I \delta_i = 1$. In practice, we initialize $u_i = 0$ for all tutors, so that $\delta_i = 1$ initially, and initialize v from a Gaussian distribution.

Training loss. While the main text presents the objective at the turn level, in implementation we group examples by (j, i) , i.e., all tutor turns from dialogue d_j^i associated with tutor i . For each turn k in such a group, we compute the margin

$$m_{j,k}^i = [\log \pi_\theta(t_{j,k}^i \mid C_i) - \log \pi_\theta(\bar{t}_{j,k} \mid C_i)] \\ - [\log \pi_\theta(t_{j,k}^i \mid C) - \log \pi_\theta(\bar{t}_{j,k} \mid C)],$$

where $C_i = A_L(\cdot) + \delta_i v$ and $C = A_L(\cdot)$. The corresponding turn loss is

$$\ell_{j,k}^i = -\log \sigma(\beta m_{j,k}^i).$$

We then average $\ell_{j,k}^i$ across turns k within a given (j, i) group, and average again across groups in the minibatch. Thus, the implementation preserves the same BiPO-style preference structure as the main objective, while stabilizing optimization by aggregating at the dialogue-tutor level.

Steering intervention. During training and inference, the intervention is applied by adding the scaled steering vector to the hidden activations at a designated transformer layer:

$$A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v.$$

During training, α is absorbed into the learned parameterization, whereas at test time we expose a global scaling factor α to control overall steering strength independently of the relative tutor-specific coefficients δ_i .

Optimization and model selection. We optimize only the steering parameters $\{v, u_1, \dots, u_I\}$, keeping the underlying language model fixed. Validation loss is computed using the same grouped objective, with cached baseline log-likelihoods for the unsteered model. The best checkpoint is selected according to validation performance, and the learned parameters are then used for test-time generation.

Inference. At test time, for a target tutor i , generation is performed with the intervention $A_L(\cdot) + \alpha \delta_i v$. This yields tutor-conditioned outputs while retaining a single shared steering direction across all tutors. The unsteered SFT model serves as the baseline condition.

Algorithm 1: Learning a steering vector and activating steering using the vector

Input: Dialogue dataset $\mathcal{D}$, SFT model π_θ , steering layer L , temperature β , training steps T

Output: Shared steering vector v , tutor-specific coefficients $\{\delta_i\}_{i=1}^I$

Pair construction:

foreach tutor i , dialogue d_j^i , tutor turn k **do**

 Construct prompt $q_{j,k}^i$ from the dialogue context

 Set chosen response to $t_{j,k}^i$

 Generate rejected response $\bar{t}_{j,k} \sim \pi_\theta(\cdot \mid q_{j,k}^i)$

 Store pair $(q_{j,k}^i, t_{j,k}^i, \bar{t}_{j,k})$

Precompute and cache $\log \pi_\theta(t_{j,k}^i \mid q_{j,k}^i)$ and $\log \pi_\theta(\bar{t}_{j,k} \mid q_{j,k}^i)$

Initialization:

Initialize $u_i \leftarrow 0$ for all i , and initialize v randomly

Set

$$\delta_i \leftarrow \frac{\exp(u_i)}{\frac{1}{I} \sum_{m=1}^I \exp(u_m)}.$$

Training:

for $t = 1, \dots, T$ **do**

 Group examples by (j, i)

foreach group (j, i) **do**

foreach turn k in d_j^i **do**

 Compute steered log-likelihoods

$$\log \pi_\theta(t_{j,k}^i \mid A_L(\cdot) + \delta_i v), \quad \log \pi_\theta(\bar{t}_{j,k} \mid A_L(\cdot) + \delta_i v)$$

 and retrieve cached baseline log-likelihoods

$$\log \pi_\theta(t_{j,k}^i \mid A_L(\cdot)), \quad \log \pi_\theta(\bar{t}_{j,k} \mid A_L(\cdot)).$$

 Form margin

$$m_{j,k}^i = \left[\log \pi_\theta(t_{j,k}^i \mid A_L + \delta_i v) - \log \pi_\theta(\bar{t}_{j,k} \mid A_L + \delta_i v) \right] - \left[\log \pi_\theta(t_{j,k}^i \mid A_L) - \log \pi_\theta(\bar{t}_{j,k} \mid A_L) \right].$$

 Compute turn loss

$$\ell_{j,k}^i = -\log \sigma(\beta m_{j,k}^i).$$

 Average $\ell_{j,k}^i$ over turns k in group (j, i)

 Average loss over groups in the minibatch and update v and $\{u_i\}$

 Recompute $\delta_i \leftarrow \exp(u_i) / (\frac{1}{I} \sum_m \exp(u_m))$

Inference:

for target tutor i **do**

 Apply activation steering

$$A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v$$

 during autoregressive generation

Table 5: Examples of our DPO dataset to train a steering vector (Part 1).

ID	Prompt	Chosen	Rejected
12024	[system] Cutting Knowledge Date: December 2023 Today Date: 26 Jul 2024 You are a tutor guiding a student through a math problem. [user] Question: Work out the area of the shape [Image: A composite shape made up of a rectangle and a triangle with some side measurements labelled] A. 68 m^2 B. 34 m^2 C. 36 m^2 D. 44 m^2 (No First Student Turn) [assistant] Hi! How can I help? [user] I dont really understand this question can you please help me with it [assistant] Of course, can you tell me how you started? Or were you just not sure at all? [user] im not sure how i started it [assistant] Ok, well I would start by splitting up the shape into a rectangle and a triangle [user] ok [assistant] Can you tell me the area of that rectangle on the left? [user] 24m squared [assistant]	Fab. Now for the triangle, the sides aren't labelled yet. So if the whole width of the shape is 10m, and the rectangle is 4m, could you work out how wide the triangle is?	Great!

Table 6: Examples of our DPO dataset to train a steering vector (Part 2).

ID	Prompt	Chosen	Rejected
1988	[system] Cutting Knowledge Date: December 2023 Today Date: 26 Jul 2024 You are a tutor guiding a student through a math problem. [user] Question: $\frac{1}{2} \times 3 =$ A. $\frac{3}{2}$ B. $3\frac{1}{2}$ C. $\frac{3}{6}$ D. $\frac{1}{6}$ First Student Turn: hi [assistant] hello 🙌 how can I help you? [user] all of this [assistant] I'm just reading the question let's think about 3 lots of $\frac{1}{2}$ so first I'm going to think about what I get if I add 3 of them together what is $\frac{1}{2} + \frac{1}{2}$? [user] 1 [assistant] great! and then add another $\frac{1}{2}$. so what is $1 + \frac{1}{2}$? [user] 1 and $\frac{1}{2}$ [assistant] 👍😊 so the answer to three times $\frac{1}{2}$ must be 1 and $\frac{1}{2}$ oh, look at the options in the question. which is the same as 1 and $\frac{1}{2}$ [user] $\frac{1}{6}$ [assistant]	no sorry, $\frac{1}{6}$ is much smaller than $\frac{1}{2}$ it's like you share something between 6 people instead of just 2 people. Each person gets less when there's 6 people to share something between instead of just 2 people sharing it have another look at the ABCD options we now know it's not D	A is $\frac{3}{2}$ 😞😞😞

Towards Just-in-Time Adaptive Feedback: Enhancing Student Learning via Knowledge-Grounded LLM

Younghun Lee[†], Amir Bralin[‡], Nobel Sanjay Rebello^{‡§}, Dan Goldwasser[†]

[†]Department of Computer Science

[‡]Department of Physics and Astronomy

[§]College of Education

Purdue University

{younghun, abralin, rebellos, dgoldwas}@purdue.edu

Abstract

Educational interventions are effective tools for enhancing student learning. While Large Language Models (LLMs) allow for generating adaptive feedback at scale, current studies lack clear methodologies for providing Just-in-Time (JiT) feedback in authentic instructional settings. In this paper, we present a framework that provides adaptive feedback by grounding LLMs with domain-specific expert knowledge. Our approach collects written reasoning logic (strategy essays) from students, analyzes potential error types based on the content of that reasoning, and delivers non-intrusive feedback designed to clarify missing or incorrect concepts. We deploy this framework in a large-scale university course ($N > 1,000$), where it improved student performance by over 80% compared to previous semesters. Lastly, we validate the framework’s pedagogical utility by analyzing the learning trajectories; we demonstrate how iterative conversations with LLM facilitate shifting one’s misconception to correct understanding.

1 Introduction

In STEM education, students are often observed to engage in “Recursive Plug-and-Chug”—also known as “Formula Hunting”—where the goal is simply to fill a slot in a formula rather than to make sense of the problem (Chi et al., 1981; Tuminaro and Redish, 2007). Existing studies indicate that while experts analyze problems based on their deep logical structure and principles, novices are often primed to surface features, leading to unsuccessful problem-solving approaches. (Mestre et al., 1993).

Strategy writing has emerged as a powerful intervention to tackle this problem. Research has shown that asking students to articulate their strategies for solving problems can improve their problem-solving skills (Leonard et al., 1996). With a combination of argumentation and appropriate prompts,

students focus more on deep structure rather than the surface features of the problem and starts solving the problem with conceptual analysis, avoiding novice, unproductive strategies (Dufresne et al., 1992; Mestre et al., 1993; Docktor et al., 2010; Rebello and Piedrahita Uruena, 2019).

Despite the benefits, strategy writing is difficult to implement at scale. The primary impediment is the feedback bottleneck. Feedback is most effective when it is integrated into the learning process through formative assessment (Hattie and Timperley, 2007), and provided prior to completion (Henderson et al., 2021). However, providing real-time, formative feedback to thousands of students is prohibitively time-consuming for human instructors. Thus, the true potential of strategy writing to enhance student learning remains largely unrealized. The advance of generative Large Language Models (LLMs) offers a promising pathway toward overcoming the scalability barrier. Recent work proposes adaptive scaffolding frameworks that utilizes LLMs to help students understand course materials (Taneja et al., 2024; Kweon et al., 2025; Kestin et al., 2025; Zhang et al., 2025). However, they lack exploration of intervention-oriented interactions as well as grounding the models with domain-specific knowledge.

In this paper, we address these challenges by exploring dimensions of adaptive feedback and implementing a framework that provides Just-in-Time interventions using knowledge-grounded LLMs. First, we collect strategy essays from students over a couple of semesters and obtain historical data that connect the essays to the types of errors students had made. Human experts then annotate feedback on a small portion of the data considering the content of the essays, as well as the error types. Using these annotated instances as few-shot examples, the LLM identifies misconceptions in the strategy essays and provide real-time, adaptive feedback. Rather than providing a ‘shortcut’ to the correct

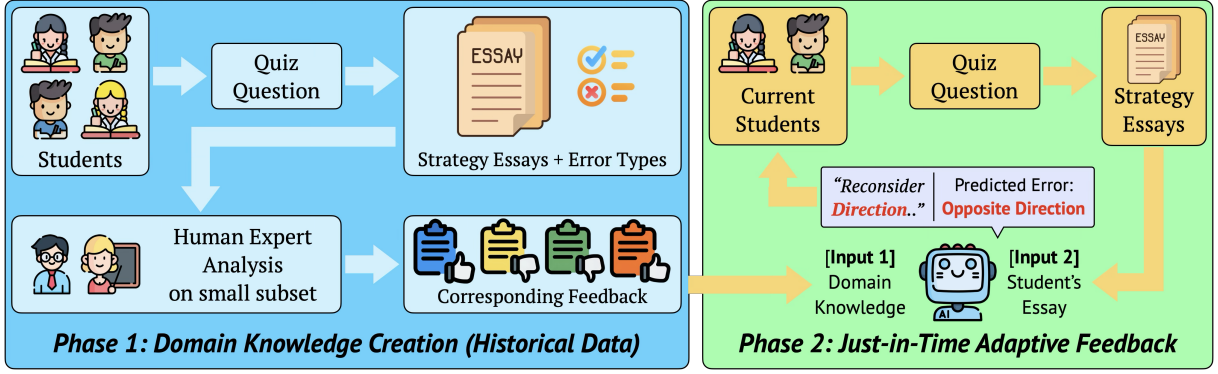


Figure 1: **Overall framework** of our Just-in-Time adaptive feedback LLM. In the first phase, we obtain domain knowledge from human experts regarding the appropriate feedback for students’ strategy essays and the types of errors they made. At deployment, we ground the LLM with this knowledge and provide adaptive feedback to the students based on their written strategy essays.

answer, the feedback directs the student’s attention to the underlying concepts and their reasoning logic (see Figure 4 and Figure 7 for examples). This ensures the intervention disrupts the ‘formula-hunting’ heuristic by letting students reflect on the problem’s deep structure. Figure 1 illustrates the overall framework we propose.

We experiment our framework on a large-scale university physics course ($N > 1,000$). Experimental results show that the overall student performance is improved when the feedback framework is deployed, by over 80% compared to previous semesters. Based on further analyses on self-reported survey and conversational instances, we argue that the framework effectively facilitates a shift from initial misconceptions to the correct understanding.

2 Related Work

Educational research has shown that feedback is one of the important drivers of learning (Wisniewski et al., 2020; Foster, 2024). Feedback can facilitate improvements in learners’ understanding and skills (Henderson et al., 2021) by narrowing the gap between actual and desired performance (Burgess et al., 2020). The effectiveness of feedback increases with the information that it contains. Previous research (Kluger and DeNisi, 1996) has also shown that moderators such as timing, specificity, and task complexity affect how learners receive and use feedback (Hattie and Clarke, 2018; Brooks et al., 2019).

Feedback is effective if it is sufficiently detailed (Price et al., 2010), usable (Winstone et al., 2017), and facilitates change (Ryan et al., 2016), such

that learners can test their new understandings (Pitt and Norton, 2017). In asynchronous and isolated online settings (Orlando, 2016), interactive dialogues can be especially useful (Wolsey, 2008) as students cannot easily interact with their peers (Furnborough and Truman, 2009) which put significant weight on the feedback comments they receive (Ortiz-Rodríguez et al., 2005).

Recent research on educational LLM applications has shifted from general assistance to specialized interventions. For instance, Phung et al. (2024) used GPT-4 as a teacher to provide non-intrusive hints but relied on LLM-simulated student agents using a weaker LLM (GPT-3.5), rather than applying it to real classroom settings. In contrast, Dai et al. (2023) and Jia et al. (2024) designed LLM feedback to apply to students, but their evaluation metrics focused on student-reported helpfulness survey and qualitative analysis of feedback content, rather than measuring actual improvements in student performance. More recent studies have shown LLM-generated feedback can help improve student performance in class. Hashmi and Rebello (2025) used an LLM-based Socratic chatbot to scaffold expert-like reasoning. Zhang et al. (2025) implemented a persona-driven multi-agent dialogue system, showing the student’s performance in problem-solving improves as they interact more with LLMs. Kestin et al. (2025) integrated an AI tutor in a college Physics course and compared student’s performance on a quiz between an AI tutor and in-class lessons. The results show that when the model is applied to a class size of 200, the AI tutor almost doubled the learning gains compared to in-class lessons.

In this paper, we introduce three novel contri-

butions. Unlike reactive chatbots that are susceptible to formula-hunting behaviors of the students, our system proactively analyzes the reasoning logic within the strategy essays that guides students to focus on deep structures in solving quiz problems. Additionally, we ground LLMs with domain-specific knowledge that is carefully annotated by human experts, which brings significant benefits compared to persona-based prompt engineering. Lastly, we provide a robust, automated intervention that remains pedagogically sound at a massive scale ($N > 1,000$).

3 Problem Formulation

3.1 Instructional Setting

Our research context is a large enrollment calculus-based physics course for engineers and physical scientists at a large U.S. Midwestern land grant university. The focus of the course is the development of problem-solving skills focused on applying key physics principles pertaining to mechanics across three units: Newton’s Laws and linear momentum, work and energy, and angular momentum. The annual enrollment of the course is around 3,300 students (1,500 in fall, 1,800 in spring), about 25% women, 10% underrepresented minorities, and 8% international students.

Each week, students complete an online quiz administered via the Learning Management System, along with an online proctoring system to maintain integrity. Students have 40 minutes to complete each online quiz. In this work, we focus on the quiz data from four consecutive semesters, from the Fall 2024 to the Spring 2026 semester.

3.2 Strategy Essays

Strategy essays refer to the student’s written reasoning logic in solving the quiz problem (Leonard et al., 1996). For one of the quiz questions, students are instructed to write a strategy essay that is at least 50 words long and does not include any numbers, symbols, or formulae. This was designed to evaluate the consistency between a student’s written reasoning logic and their performance on the quizzes. To maintain the quality of the essays, the instructors offer extra credit¹ for writing a strategy essay. In the Fall 2024 semester, for example, we gathered 11,948 essays from 1,418 students from a total of 11 quizzes.

¹Extra credits equal to 10% of the total points on the quiz

3.3 Desired Properties of LLM Feedback

LLM feedback refers to the LLM-generated text that is given to each student. The primary purpose of LLM feedback is to help students better understand the course materials related to the quiz problem and solve it correctly.

One of the most critical constraints in providing feedback is **to facilitate the student’s own problem-solving process**. The LLM should not disclose a direct solution or correct answers to the quiz problems when generating feedback to students. Another aspect we consider in designing the feedback is **adaptiveness**; rather than generating universal feedback, it is conditioned on student-specific features such as their in-class performance (e.g. pre-semester assessment scores, midterm exam scores, other quiz scores, etc.) and their strategy essays. Lastly, we focus on **grounding LLMs with domain-specific knowledge**. General-purpose LLMs often exhibit unreliable performance in understanding college-level STEM courses that require multi-step logic (Arora et al., 2023; Pang et al., 2025). To minimize the knowledge gap, we distill domain-specific knowledge from experts (i.e. instructors) and ground LLM generations with it.

3.4 Post-Hoc vs. Just-in-Time (JiT)

LLM feedback can be presented to students in two different ways: after students finish solving the quiz (post-hoc) and in real-time while they solve it (JiT). These two methods serve different benefits. In this paper, we utilize LLM post-hoc feedback to mainly survey the student’s preference for the feedback style. Students’ preference patterns ultimately help identify how LLM feedback needs to be generated. We consider LLM JiT feedback as a means of real-time interventions which help students solve the quiz problem.

4 Designing Adaptive Feedback

4.1 Student Preference Survey

Existing studies suggest the necessity of adaptive guidance that aligns with the student’s level of knowledge; unlike the way novice students perform better with detailed explanations, more experienced learners find it redundant and even show negative consequences (Kalyuga, 2007; Walker et al., 2012; Albacete et al., 2019). Since there is limited work that verifies this finding with LLM-based feedback, we conduct a series of surveys to find out whether

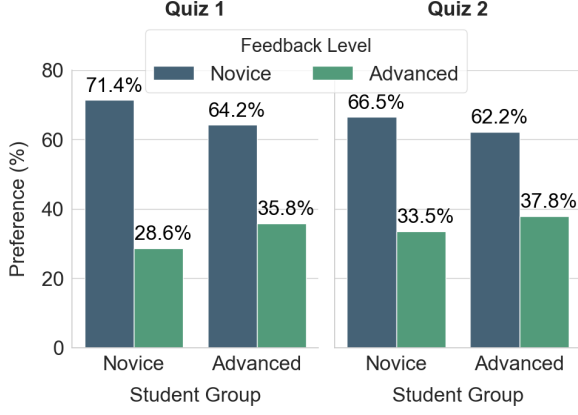


Figure 2: **Survey results regarding preference** between novice and advanced feedback in two quizzes. Regardless of their level of knowledge, students prefer LLM feedback that is targeted to the novice group.

students with different knowledge levels show a contrasting preference for the complexity of LLM feedback.

We provide post-hoc LLM feedback to the students for two quizzes in the Fall 2025 semester. Students are asked to write a short strategy essay that they employ to solve the problems. After having strategy essays from the students, we prompt LLMs to generate two different versions of the feedback, one for the novice-level and the other for the advanced-level. LLMs are instructed to differentiate not only linguistic styles such as tone and vocabulary, but also how feedback is framed with respect to the focus and goal². Students are instructed to participate in the survey on a separate webpage showing two feedback texts, one targeting the novice-level and the other for the advanced-level, and are asked to indicate which version they prefer. The presentation order of the feedback was randomized to mitigate potential order effects and minimize response bias. Figure 10 shows the survey webpage (see Appendix A).

To find the relevance between the student’s level of knowledge and the complexity of the LLM feedback, we categorize students into two buckets, novice and advanced. Students were categorized as novice if they performed below average for all relevant assessments; this includes a pre-semester assessment³, mid-term exam, and relevant previous quizzes. In contrast, students who consistently exceeded the average score for all relevant assess-

²We used GPT-5.1 to generate post-hoc feedback. Detailed settings for prompting are described in Appendix A

³Energy and Momentum Conceptual Survey (Singh and Rosengrant, 2003)

Quiz Problem

Question
 What is the horizontal component of the force that top block exerts on the bottom block?

Answer Options

A. $-m_2 \frac{\vec{F}}{m_1 + m_2}$	B. $m_2 \frac{\vec{F}}{m_1 + m_2}$
C. $-m_1 \frac{\vec{F}}{m_1 + m_2}$	D. $m_1 \frac{\vec{F}}{m_1 + m_2}$

Figure 3: **Example quiz problem.** While A is correct, other options map to common errors: B represents a direction error (misunderstanding force directions); C represents a position error (incorrect mass/location consideration); and D represents confusion involving both concepts (position-direction error).

ments were categorized as advanced.

Figure 2 shows the results. Although the portion of students who favor advanced-level feedback is larger among students with advanced knowledge, both novice and advanced groups prefer feedback that is targeted to the novice group. This implies that LLMs are inadequate to generate advanced feedback that is pedagogically effective, suggesting that simply instructing LLMs to increase vocabulary complexity and reduce scaffolding may hinder rather than help the learning process.

4.2 Just-in-Time Intervention

Using insights from the survey results, we focus on a different aspect of adaptive LLM feedback for JiT intervention: **error types**.

All incorrect multiple-choice options in the quizzes are designed to elicit specific error types. Consider an example quiz in Figure 3. The question asks to find the force that the top block exerts on the bottom block, given a certain amount of horizontal force, $\vec{F}$, is applied to the bottom block and the two blocks move together. The correct answer is multiplying the mass of the top block by acceleration which is the opposite direction to the $\vec{F}$. Common errors can be made by not considering that the direction has to be opposite (B. direction

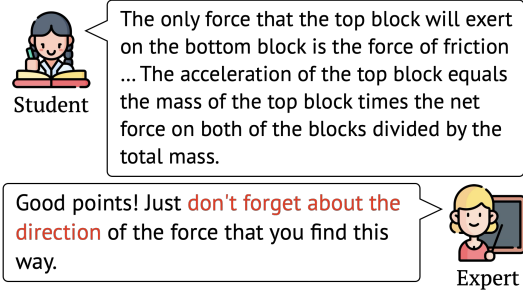


Figure 4: Example of a strategy essay and annotated feedback. Although the student in this example got the correct answer, the feedback mentions the importance of considering direction as it was missing in the essay.

error), or by multiplying the mass of another object (C. position error), or by having confusion involving both concepts (D. position-direction error). We posit that LLMs offer effective interventions by predicting a student’s likely error type and providing adaptive feedback that guides them to reconsider those aspects in their final answer.

4.3 Grounding LLM Feedback with Domain Knowledge

As stated in 3.3, LLMs need to be grounded with domain-specific knowledge to provide more effective feedback with a better understanding of the concepts. We integrate human-expert knowledge into LLMs’ feedback generation process.

Using historical data from previous semesters, we collected students’ strategy essays and their corresponding answer choices, enabling us to associate written reasoning with particular error types. Teaching experts (i.e. faculty and instructors) then manually wrote feedback tailored to the content of these essays and error types.

Such nuanced feedback annotation is naturally a challenging task for LLMs. For example, students who selected the correct answer might still write imperfect strategy essays by not explicitly mentioning the force direction. Conversely, a semantically similar essay written by another student might lead to the direction error. Human experts are more competent in capturing these subtle differences between strategy essays and answer choices. Figure 4 illustrates an example of the annotation process, where experts identify missing concepts regardless of whether the student’s final answer was correct.

Once the feedback annotation is completed for a small representative set (50 examples), they serve as the few-shot examples for in-context learning, enabling the real-time classification of error types

Method	Accuracy	Macro F1
Fine-tuned Classifier	50.00 $\pm$ 0.001	28.39 $\pm$ 0.001
Zero-shot LLM	21.00 $\pm$ 0.004	12.74 $\pm$ 0.006
Zero-shot LLM w/ Secondary	37.00 $\pm$ 0.003	28.34 $\pm$ 0.005
Few-shot LLM	42.42 $\pm$ 0.001	33.50 $\pm$ 0.002
Few-shot LLM w/ Secondary	60.61 $\pm$ 0.002	54.24 $\pm$ 0.003

Table 1: Classification performance of different models in predicting error types (i.e. correct, direction, position, position-direction) given strategy essay. LLMs perform better when they are prompted with few-shot examples and Chain-of-Thought reasoning while considering secondary label.

using student’s strategy essays. Recognizing that strategy essays are often non-trivial to categorize into potential error types, we move beyond single-label prediction; we prompt the LLMs to identify both the primary and secondary most likely labels. The model then produces feedback via Chain-of-Thought reasoning (Wei et al., 2022), conditioned on this expanded label space to ensure more robust instructional support. Detailed explanations regarding prompts and JiT intervention framework are provided in Appendix C.

5 Experiments and Results

5.1 Error Type Predictions

In this experiment, we measure the accuracy of error type prediction across different models. The task is formulated as a multi-class classification where the model gets a strategy essay as input and predicts one of four classes: correct, or one of the three error types—direction, position, and position-direction. Table 1 describes the accuracy and macro F1 score of the models.

As a baseline, we first implement a sentence classifier using a pre-trained BERT-based checkpoint using the quiz data from Fall 2024 semester. Since the number of data instances is small (1.1K), the trained model shows zero accuracy on labels with insufficient amount of training instances, results in a very low macro F1 score.

As the number of training data instances is limited, we use high-performance LLMs to classify the error types without fine-tuning. We try prompting the LLMs with zero-shot and few-shot settings (3 examples per label). Although adding few-shot examples improve the performance, the accuracy and macro F1 scores are not strong enough. After we prompt the model to consider the secondary most likely labels, the performance increases signif-

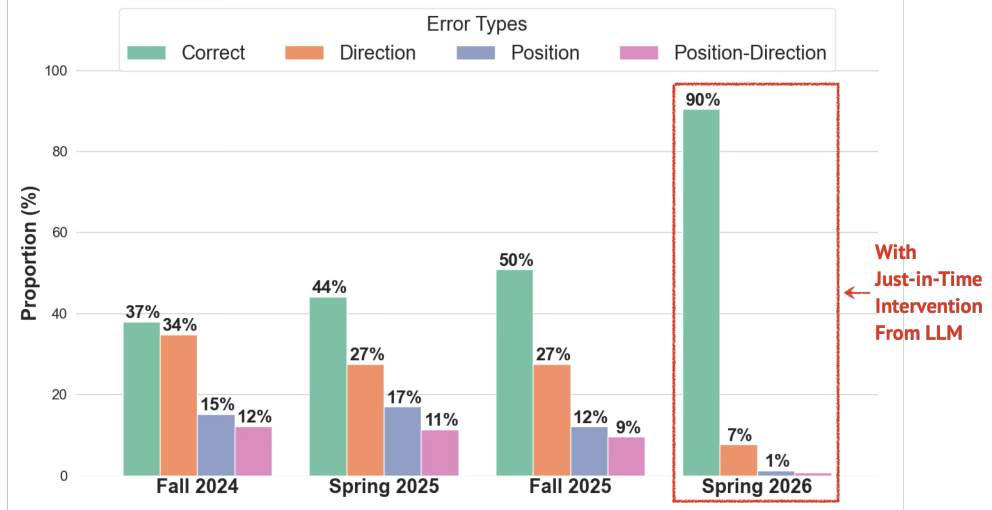


Figure 5: Student’s performance in the same quiz question over four semesters. When the students solved the problem with the LLMs providing JiT adaptive intervention, the overall portion of correct students boosted significantly.

icantly, implying that the LLM can generate more robust feedback with expanded label space. Detailed explanations regarding prediction models are described in Appendix B.

5.2 Quiz with JiT Interventions

We administered JiT interventions during a specific quiz in the Spring 2026 semester. The problem and its corresponding multiple-choice options are illustrated in Figure 3. Students accessed the quiz through their standard Learning Management System, with the LLM feedback interface integrated directly into the problem description. A comprehensive view of the user interface and the interaction flow is provided in Figure 13 (see Appendix C). Upon engaging with the system, students were informed that the LLM would not provide direct solutions or answers. Instead, they were instructed to use the framework to get advice on specific concepts requiring further attention, based on the content of their strategy essays.

5.3 Student Performance Improvements

To measure the effectiveness of LLM-generated feedback, we compare student performance on the same quiz question over four consecutive semesters. As shown in Figure 5, more than half of the class in previous semesters made errors. In contrast, the error rate drops to less than 10% with the help of JiT LLM intervention.

Notably, the feedback remains highly effective despite sub-optimal performance of LLMs in predicting error types. We attribute this to domain-

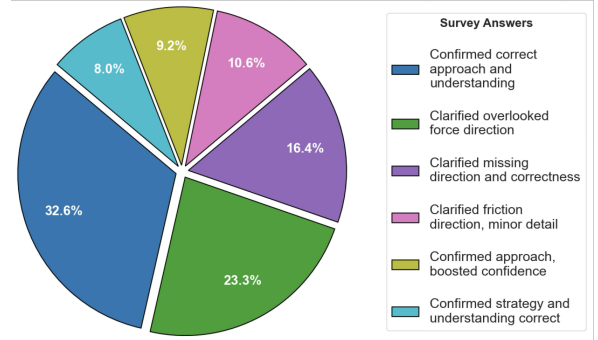


Figure 6: Survey results regarding why LLM’s feedback was helpful. Aside from confirmation, students indicate that LLM was useful in clarifying overlooked / missing aspects in the essays

knowledge grounding, which incorporated expert-generated feedback that addresses potentially missing or incorrect concepts regardless of the predicted error class.

After students solve the quiz with LLM feedback, we ask whether the feedback was helpful and why. Approximately 77.59% of the students reported the feedback was helpful; their qualitative responses are clustered using HDBSCAN, and shown in Figure 6.

There are two major themes emerged from the clusters. Around half of the students valued the feedback for concept verification and confirmation, while the other half stated that the LLMs clarified overlooked or missing items in their strategy essays. This illustrates the adaptiveness of JiT LLM feedback; for students with a correct understanding,

Metric	Value
Total instances	1,042
Conversational instances	209 (20.05%)
Mean # conv. turns (std)	2.77 (1.52)
Min/Max # conv. turns	2 / 14
Conv. turn dist. skewness	3.57
% of Model Prediction as Correct	
Turn 1 (Initial essay)	42.79%
Last turn (Final essay)	71.64%

Table 2: Statistics of conversational instances. The skewness of the number of conversation turn distributions is computed by Fisher-Pearson coefficient of skewness where the value greater than 2 is normally considered extremely skewed.

the LLM provides essential validation, whereas for those prone to errors, it guides them rethink or pay more attention to specific concepts.

5.4 Learning through Conversations

One of the emergent behaviors of students in interacting with feedback-generating LLM is that they have conversations with the LLMs to improve their understanding. This observation is impressive considering that students are not directed or incentivized to iteratively chat with the LLMs. Students are notified as solving the quiz with LLMs will give them a bonus credit regardless of the correctness of their answers. Nevertheless, there are around 20% of the students who voluntarily spent more time interacting with the LLM so that they could improve their learning experience. Basic statistics of the conversational instances are reported in Table 2, and more detailed statistics are reported in Appendix D.

Figure 7 shows an example of conversation between a student and LLM. As the student receives LLM’s feedback that tackles missing or incorrect points in the strategy essay, they add these components to their strategy essay. By continuing conversations, students incrementally add missing points and gain more comprehensive understandings. Table 2 illustrates such instances: while only 43% of students’ initial essays were predicted as correct, the portion boosted to 72% after having conversations.

We visualize learning trajectories by analyzing the sequence of conversation turns with the LLM. This involves tracking the intermediate strategy essays and their corresponding predicted error types throughout the interaction. Figure 8 illustrates how students modify their strategy essays and how the

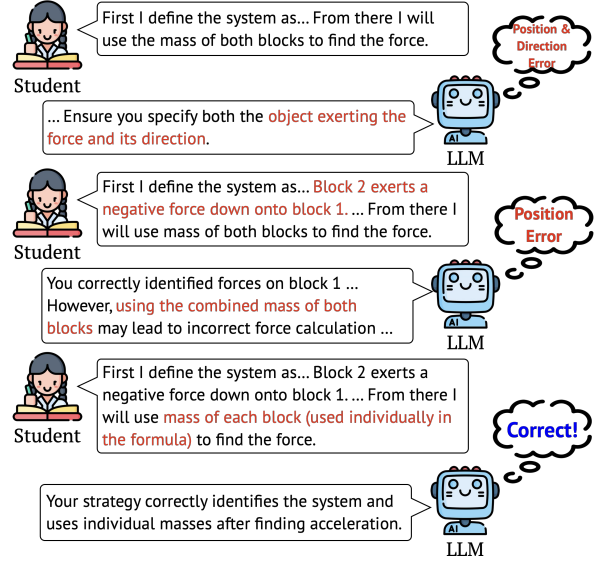


Figure 7: **Example Conversation** of a student modifying their strategy essays with the help of LLM. LLM generates feedback based on its error type prediction, and informs different aspects as the student fixes errors one by one.

predicted error types change while they have conversations with LLMs. The figure represents a subset where students’ strategy essays are initially classified as having a direction error.

As conversation proceeds, students keep modifying their essays by applying LLM’s advice, and there exists multiple learning trajectories based on how they improve their essays. In the end, approximately 80% of the students corrected their essays and the LLM classified them as ‘correct’. Other students, even though their final version of essays are not classified as ‘correct’, they solved the quiz correctly. We assume that this is the case where students skip the final validation step after having their perspectives corrected. Overall, 91.43% of the students (32 out of 35) solved the quiz correctly by iteratively getting feedback from LLM and developing their understandings.

We conducted a further analysis to validate the hypothesis that students who engage more actively with the LLM feedback improve their strategy essays better. We defined “activeness” through two metrics: reflective latency (the time in seconds between conversation turns) and degree of revision (the change in word count between consecutive essay versions). Specifically, we posit that spending more time refining an essay or making more substantial text changes indicates a more active adaptation of the feedback.

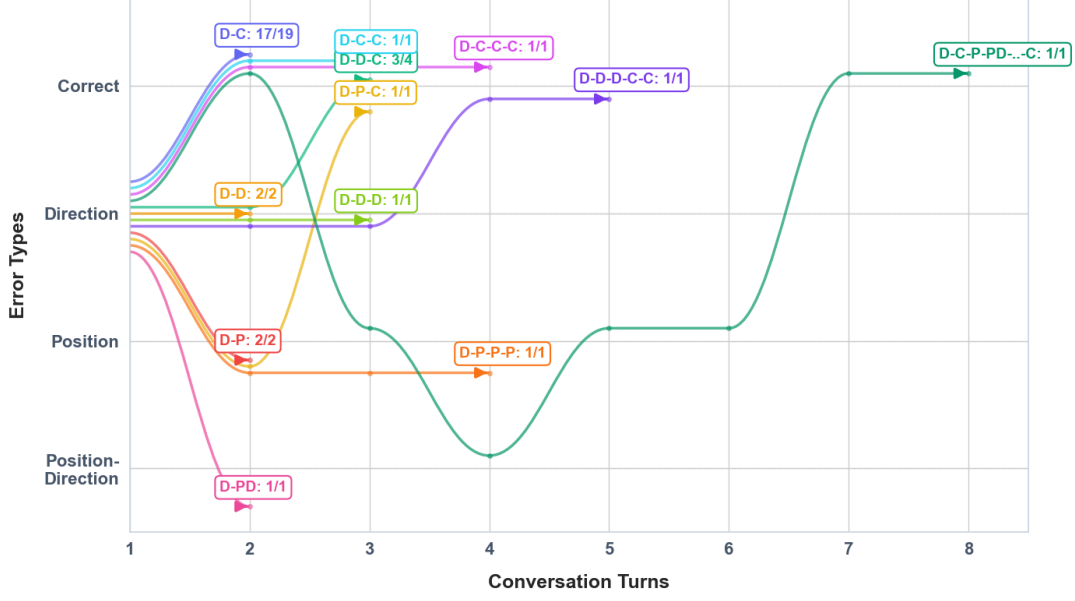


Figure 8: **Learning Trajectories** for students starting with Direction error. Each trace represents a sequence of essay error types as predicted by the feedback-LLM at each conversation turn. Terminal markers are labeled with the specific Path Sequence and the Path Outcome. For instance, ‘D-C: 17/19’ indicates that out of 19 students whose essays the LLM classified as ‘Direction error’ and then ‘Correct’, 17 got correct answers in the end.

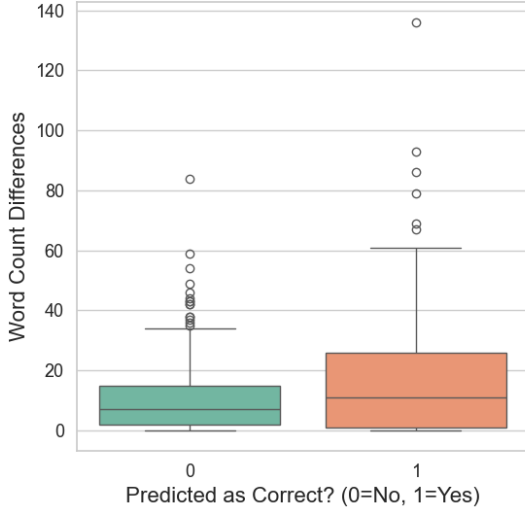


Figure 9: **Correlation between word count differences and essay correctness.** We compute the correlation between the word count differences in strategy essays between each conversation utterances and the correctness of essay. When the students make more changes to their essays, they are more likely getting their essays correct.

To evaluate these hypotheses, we compute the Pearson correlation coefficients. The results indicate a negligible relationship between time spent and essay correctness ($r = 0.06$, $p = 0.27$). In contrast, the correlation between word count dif-

ference and essay correctness is statistically significant ($r = 0.18$, $p < 0.001$). Figure 9 illustrates how word count differences are distributed differently for the essays that are classified as correct or not. These findings suggest that simply spending more time on the interface is insufficient; rather, the degree of revision is the primary factor of success. This implies that while the LLM provides the necessary feedback, the likelihood of achieving a correct understanding depends heavily on the student’s willingness to actively reformulate their reasoning.

6 Conclusion

In this paper, we explore how LLMs can be utilized efficiently in higher education through Just-in-Time (JiT) adaptive interventions. By analyzing strategy essays from students, our framework identifies potential errors and provides adaptive feedback grounded in domain knowledge. Experimental results show that our framework significantly improves the overall student performance by providing validation and guiding re-evaluation of incorrect concepts. Qualitative analysis further demonstrates that students engage in conversations with LLMs to improve their understanding, highlighting the potential for LLMs to serve as sophisticated pedagogical tools.

Limitations

One of the limitations of this work is the sub-optimal accuracy of the error type classification. Predicting error types solely from strategy essays is a non-trivial task even for humans. Students often omit important details, whether due to oversight or lack of conceptual understanding. LLM feedback falls into a catastrophic failing mode when the LLM incorrectly classifies an essay as ‘correct’ and does not provide necessary interventions as a result of that. While we mitigate this risk by considering second most-likely label, the reliability and robustness of the framework can be improved with better classification. In future iterations, we could improve it by using relevant metadata (e.g. student’s level of knowledge, tone and vocabulary used in the essay, etc.), or by enforcing a more structured essay format to ensure all necessary concepts are articulated.

Another limitation is the small number of JiT interventions, as the framework was applied to only a single quiz during the semester. Although we observe significant performance improvements with the LLM feedback, a broader deployment across multiple topics and quiz types is necessary to validate these findings. Expanding the scope would allow for a more granular analysis of which specific subject matters or problem types benefit most from LLM-based interventions, providing deeper insights in how LLM feedback should be designed.

Our student preference survey shows that while the LLM is highly effective for novice-level feedback, it struggles to generate feedback that is more suitable to students with advanced knowledge; students with over-the-average performance still preferred LLM feedback that is targeted to novice-level over the advanced-level. This opens a new direction for designing LLM feedback in future studies to provide minimal interventions considering the knowledge level of students.

From a technical point of view, approximately 4% of the students experienced server failures due to the high volume of synchronous API calls, mainly due to the size of the course being very large ($N > 1,000$). To make the framework more scalable, future implementations need to address rate-limiting and asynchronous processing. Furthermore, the current LLM lacks a memory function that help efficient conversations; the current version processes updated strategy essays in isolation without the context from previous versions. In

future work, we aim to incorporate student-specific memory and conversation history that reduces the number of turns required for a student to reach a correct understanding.

Ethics Consideration

This research was declared ‘Exempt’ under IRB protocol number IRB-2023-391. The participant pool consisted of undergraduate students enrolled in a large-scale introductory Physics course. All personally identifiable information, including student names and university IDs, was removed from the data before being processed by the LLMs. Student essays used for qualitative analysis in this paper have been further anonymized, truncated, and paraphrased.

Our framework contains risks when generating feedback to students, primarily because of hallucinations and incorrect classification of error types. To mitigate these risks, the LLM was grounded with expert-annotated domain knowledge as well as its secondary predictions. Furthermore, we prioritized equity of treatment and algorithmic fairness throughout the study. The LLM was prompted using a standardized, invariant template for all participants, ensuring that the feedback was controlled solely by the student’s essay rather than any external variables such as student’s personal information or prior performance levels. By maintaining a uniform prompting strategy, we ensured that no specific group of students was prioritized or discriminated against, providing consistent pedagogical support to the entire students.

We disclose that LLMs are used for checking grammatical errors and exploring paraphrases. All core research contributions including motivation, experiment designs, and results analyses were developed by the authors.

Acknowledgments

This research is supported in part by U.S. National Science Foundation grant 2300645. Opinions expressed are of the authors and not of the Foundation.

References

Patricia Albacete, Pamela Jordan, Sandra Katz, Irene-Angelica Chounta, and Bruce M McLaren. 2019. The impact of student model updates on contingent scaffolding in a natural-language tutoring system. In

- International conference on artificial intelligence in education*, pages 37–47. Springer.
- Daman Arora, Himanshu Singh, and 1 others. 2023. Have llms advanced enough? a challenging problem solving benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7527–7543.
- Cameron Brooks, Annemaree Carroll, Robyn M Gillies, and John Hattie. 2019. A matrix of feedback for learning. *Australian Journal of Teacher Education (Online)*, 44(4):14–32.
- Annette Burgess, Christie van Diggele, Chris Roberts, and Craig Mellis. 2020. Feedback in the clinical setting. *BMC medical education*, 20(Suppl 2):460.
- Micheline TH Chi, Paul J Feltovich, and Robert Glaser. 1981. Categorization and representation of physics problems by experts and novices. *Cognitive science*, 5(2):121–152.
- Wei Dai, Jionghao Lin, Hua Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gašević, and Guanliang Chen. 2023. Can large language models provide feedback to students? a case study on chatgpt. In *2023 IEEE international conference on advanced learning technologies (ICALT)*, pages 323–325. IEEE.
- Jennifer L Docktor, Natalie E Strand, José P Mestre, and Brian H Ross. 2010. A conceptual approach to physics problem solving. In *AIP Conference Proceedings*, volume 1289, pages 137–140. American Institute of Physics.
- Robert J Dufresne, William J Gerace, Pamela Thibodeau Hardiman, and Jose P Mestre. 1992. Constraining novices to perform expertlike problem analyses: Effects on schema acquisition. *The Journal of the Learning Sciences*, 2(3):307–331.
- H Foster. 2024. The impact of formative assessment on student learning outcomes: A meta-analytical review. *Academy of Educational Leadership Journal*, 28(S1):1–3.
- Concha Furnborough and Mike Truman. 2009. Adult beginner distance language learner perceptions and use of assignment feedback. *Distance Education*, 30(3):399–418.
- Syed Furqan Abbas Hashmi and N. Sanjay Rebello. 2025. Analyzing undergraduate problem-solving in physics through interaction with an ai chatbot. In *Physics Education Research Conference 2025, PER Conference*, pages 184–189, Washington, DC.
- John Hattie and Shirley Clarke. 2018. *Visible learning: feedback*. Routledge.
- John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.
- Michael Henderson, Tracii Ryan, David Boud, Phillip Dawson, Michael Phillips, Elizabeth Molloy, and Paige Mahoney. 2021. The usefulness of feedback. *Active Learning in Higher Education*, 22(3):229–243.
- Qinjin Jia, Jialin Cui, Haoze Du, Parvez Rashid, Ruijie Xi, Ruochi Li, and Edward Gehringer. 2024. Llm-generated feedback in real classes and beyond: Perspectives from students and instructors. In *Proceedings of the 17th international conference on educational data mining*, pages 862–867.
- Slava Kalyuga. 2007. Expertise reversal effect and its implications for learner-tailored instruction. *Educational psychology review*, 19(4):509–539.
- Greg Kestin, Kelly Miller, Anna Klales, Timothy Milbourne, and Gregorio Ponti. 2025. Ai tutoring outperforms in-class active learning: An rct introducing a novel research-based design in an authentic educational setting. *Scientific Reports*, 15(1):17458.
- Avraham N Kluger and Angelo DeNisi. 1996. The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological bulletin*, 119(2):254.
- Sunjun Kweon, Sooyohn Nam, Hyunseung Lim, Hwajung Hong, and Edward Choi. 2025. A large-scale real-world evaluation of an llm-based virtual teaching assistant. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 850–864.
- William J Leonard, Robert J Dufresne, and Jose P Mestre. 1996. Using qualitative problem-solving strategies to highlight the role of conceptual knowledge in solving problems. *American Journal of Physics*, 64(12):1495–1503.
- Jose P Mestre, Robert J Dufresne, William J Gerace, Pamela T Hardiman, and Jerold S Touger. 1993. Promoting skilled problem-solving behavior among beginning physics students. *Journal of research in science teaching*, 30(3):303–317.
- John Orlando. 2016. A comparison of text, voice, and screencasting feedback to online students. *American Journal of Distance Education*, 30(3):156–166.
- Madeline Ortiz-Rodríguez, Ricky W Telg, Tracy Irani, T Grady Roberts, and Emily Rhoades. 2005. College students’ perceptions of quality in distance education: The importance of communication. *Quarterly Review of Distance Education*, 6(2):97–105.
- Xinyu Pang, Ruixin Hong, Zhanke Zhou, Fangrui Lv, Xinwei Yang, Zhilong Liang, Bo Han, and Changshui Zhang. 2025. Physics reasoner: Knowledge-augmented reasoning for solving physics problems with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11274–11289.

- Tung Phung, Victor-Alexandru Pădurean, Anjali Singh, Christopher Brooks, José Cambronero, Sumit Gulwani, Adish Singla, and Gustavo Soares. 2024. Automating human tutor-style programming feedback: Leveraging gpt-4 tutor model for hint generation and gpt-3.5 student model for hint validation. In *Proceedings of the 14th learning analytics and knowledge conference*, pages 12–23.
- Edd Pitt and Lin Norton. 2017. ‘now that’s the feedback i want!’ students’ reactions to feedback on graded work and what they do with it. *Assessment & Evaluation in Higher Education*, 42(4):499–516.
- Margaret Price, Karen Handley, Jill Millar, and Berry O’donovan. 2010. Feedback: all that effort, but what is the effect? *Assessment & Evaluation in Higher Education*, 35(3):277–289.
- Carina M Rebello and Yuri Piedrahita Uruena. 2019. Using a hybrid of argumentation and problem solving prompts to facilitate undergraduates’ problem solving performance and confidence. In *The 13th Conference of the European Science Education Research Association (ESERA)*.
- Tracii Ryan, Michael Henderson, and Michael Phillips. 2016. Written feedback doesn’t make sense’: Enhancing assessment feedback using technologies. In *International conference of the Australian Association for Research in Education*, pages 1–11.
- Chandralekha Singh and David Rosengrant. 2003. Multiple-choice test of energy and momentum concepts. *Am. J. Phys.*, 71(6):607–617.
- Karan Taneja, Pratyusha Maiti, Sandeep Kakar, Pranav Guruprasad, Sanjeev Rao, and Ashok K Goel. 2024. Jill watson: A virtual teaching assistant powered by chatgpt. In *International Conference on Artificial Intelligence in Education*, pages 324–337. Springer.
- Jonathan Tuminaro and Edward F Redish. 2007. Elements of a cognitive model of physics problem solving: Epistemic games. *Physical Review Special Topics—Physics Education Research*, 3(2):020101.
- Erin Walker, Nikol Rummel, Sean Walker, and Kenneth R Koedinger. 2012. Noticing relevant feedback improves learning in an intelligent tutoring system for peer tutoring. In *International Conference on Intelligent Tutoring Systems*, pages 222–232. Springer.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Naomi E Winstone, Robert A Nash, James Rowntree, and Michael Parker. 2017. ‘it’d be useful, but i wouldn’t use it’: barriers to university students’ feedback seeking and recipience. *Studies in Higher Education*, 42(11):2026–2041.
- Benedikt Wisniewski, Klaus Zierer, and John Hattie. 2020. The power of feedback revisited: A meta-analysis of educational feedback research. *Frontiers in psychology*, 10:487662.
- Thomas Wolsey. 2008. Efficacy of instructor feedback on written work in an online program. In *International Journal on E-learning*, volume 7, pages 311–329. Association for the Advancement of Computing in Education (AACE).
- Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, and 1 others. 2025. Simulating classroom education with llm-empowered agents. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10364–10379.

A Feedback Preference Survey

We survey student’s LLM feedback preference in the Fall 2025 semester for two quizzes. Students are instructed that they will get extra credits for participating in the survey, and the number of participants for each iteration was 1,074 and 1,160. The two options for feedback is generated with the prompt described in Figure 11; it first evaluates the student’s strategy essays using expert-annotated rubrics, then generate two different versions of feedback using gpt-5.1, one for novice and the other for advanced.

Once the feedback generation is completed, we hosted a separate survey webpage from the student’s standard Learning Management System as shown in Figure 10. The two options are randomly swapped to minimize any potential biases. Students are instructed to read the quiz problem and essay they authored to remind themselves of the context, and choose one of the feedback texts they prefer with a set of pre-defined reasons (students can choose multiple reasons). For both surveys, the most selected reason was “Helps me better understand the concept”, followed by “Better explains the errors in my strategy” and “Aligns better with my level of knowledge”.

B Error Type Prediction Model Details

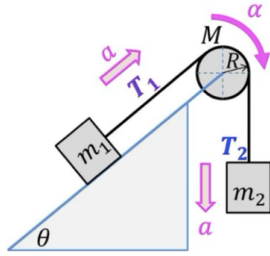
For fine-tuned classifiers, we randomly split the data into 80-10-10 ratio for train-validation-test sets. We use DeBERTa-v3-small model with learning rate of $2e-5$ and batch size of 16. The model is fine-tuned and tested on a single NVIDIA A100 GPU with 80GB memory, takes around 10 minutes to fine-tune 5 epochs.

Quiz 11 Survey. Which Feedback Do You Prefer?

Review the feedback options for your strategy essay and choose an option which you find most helpful.

Quiz Question

In the Atwood's Machine on the inclined plane as shown, the rope does not slip over the pulley with bearings that have negligible friction. The masses of the two blocks shown are known. The inclined plane makes a known angle with the horizontal. The coefficient of friction between the block m_1 and the inclined plane is known. The pulley is a disk of known mass and known radius. What is the linear acceleration, a , of each block?



Your Essay

To find the acceleration of each block, I must first set up the Momentum Principle for each block, which states that the net force acting on the block is equal to its mass times its acceleration. However, since that gives me only two equations with more than two unknowns, I must also use the Angular Momentum Principle for the pulley, which states that the net torque on the pulley is equal to its moment of inertia times its angular acceleration. I can write the angular acceleration in terms of the linear acceleration of each block since the rope is not slipping. Then I will plug in known variables, and use substitution to solve the system of equations for the acceleration of each block.

Feedback Options

Feedback A

You've identified the correct governing principles and the no-slip kinematic relation, but your plan is missing some critical structural details—most notably the need to treat the tensions on either side of the pulley as distinct and to explicitly include friction and weight components in your force balances. Before plugging in variables, try to write three fully specified equations: two linear momentum equations with properly resolved components (including friction for m_1 and weight components along the incline) and one rotational equation with torque from the two different tensions and the correct disk moment of inertia. As you refine your strategy, ask yourself: where exactly does the torque on the pulley come from if the tensions were equal, and how does the frictional interaction on the incline qualitatively affect the sign and magnitude of the acceleration you expect?

Feedback B

You've made a very strong start by correctly naming the two big ideas: using Newton's Second Law (momentum principle) for the blocks and rotational Newton's Second Law (angular momentum principle) for the pulley. One important next step is to be more explicit about the forces: write separate force equations for each block that include gravity (weight), friction on the incline, and the tension pulling each block, and also remember that the tension on each side of the pulley can be different so that there is a net torque on the pulley. It's great that you know to connect angular acceleration to linear acceleration with the no-slip condition; if you now carefully list all forces and write three equations (one for each block and one for the pulley), you'll have everything you need to solve for the acceleration with confidence.

1. Choose better feedback

☐ A ☒ B

2. Why do you prefer this feedback?

Please select all that apply. At least one option is required.

- ☐ Helps me better understand the concept
- ☐ Better explains the errors in my strategy
- ☐ Aligns better with my level of knowledge
- ☐ Other

Please specify

Submit

Figure 10: Survey webpage. Students answer two questions; choosing better feedback and the reason why.

For LLM-based inference, we used the deepseek-r1:70B checkpoint, hosted on-premise at the authors' institution, to make sure the student's essay data is not shared outside.

All fine-tuning and inferences are performed with 2 trials and the average score was reported with the confidence interval in Table 1.

C JiT Feedback Implementation

When the LLM is deployed for generating JiT feedback, we host a web app using Flask API, which is later embedded to the Learning Management System that students use for the course. The web app is a simple text box where students put their strategy essays and clicks a button to get the feedback from the LLM. In the back-end, we use pre-defined few-shot examples with Chain-of-Thought reasoning so that the model considers primary and secondary class of the strategy essay. The example of an actual prompt and the quiz page students have accessed is shown in Figure 12 and Figure 13.

D Conversation Analysis

From \ To	Correct	Position	Direction	Pos-Dir
Correct	0.6797	0.1328	0.1250	0.0625
Position	0.3491	0.3585	0.1415	0.1509
Direction	0.5278	0.1250	0.2917	0.0556
Pos-Dir	0.3750	0.2969	0.0781	0.2500

Table 3: Transition probabilities between error types across conversation turns

In this section, we report additional analyses on conversational instances. Table 3 shows the transition probabilities between conversation turns. When the student's strategy essay at the i -th conversation turn is predicted as correct, it is still going to be classified as the same label with the 67.97% chances, but there exists slightly more than 30% chances the essay is classified as one of the error types. While majority of the strategy essays with direction and position-direction errors are likely developed into correct strategy, essays with position error is more likely going to stay at the same error type. This implies that LLM's ability to guide students avoid errors differ based on the types.

Prompt Sample: Generating Post-hoc Feedback for Preference Survey

You are an expert College Teaching Assistant and Pedagogical Specialist. Your goal is to provide feedback on a student's "Strategy Essay" regarding a quiz problem. First, analyze the student's written strategy to infer their Knowledge Level based on the complexity and accuracy of their writing. Then, regardless of their actual level, generate two distinct versions of feedback (one adapted for a "Novice" and one for an "Advanced" student) to provide options for instruction. Generate each feedback in 3 sentences.

Input Data Provided:

1. The Quiz Problem: The specific question the student attempted to solve.
2. Student's Strategy Essay: The student's written explanation of how they approached the problem.
3. Expert's Strategy Essay: A sample strategy essay written by an expert, that can serve as a rubrics for evaluating student's essay

Instructions for Adaptation:

- If Knowledge Level is "NOVICE":
 - Tone: Highly encouraging, supportive, and patient. Use the "Sandwich Method" (Praise -> Gentle Correction -> Praise).
 - Vocabulary: Avoid jargon where possible. If technical terms are used, briefly define them or use analogies.
 - Focus: Focus on identifying misconceptions and solidifying the fundamental concept.
 - Scaffolding: Break down the next steps into small, manageable pieces. Guide them toward the correct starting point.
 - Goal: To build confidence and ensure they understand the basic mechanism of the problem.
- If Knowledge Level is "ADVANCED":
 - Tone: Professional, peer-to-peer, challenging, and concise.
 - Vocabulary: Use precise, high-level academic/technical terminology appropriate for the field.
 - Focus: Focus on optimization, edge cases, and connecting this problem to broader course concepts. Point out subtle logic flaws or inefficiencies.
 - Scaffolding: Do not hand-hold. Ask Socratic questions that force the student to re-evaluate their own logic.
 - Goal: To refine their reasoning, improve efficiency, and deepen metacognition.

Feedback Structure (Output Format) Please output the response in the following json format:

```
{ "Essay_Evaluation": "1-2 sentences summarizing their approach based on the expert sample",  
  "Inferred_Level": "Either Novice or Advanced",  
  "Feedback": {  
    "Novice": "The core feedback paragraph. Apply the Instructions for Adaptation logic for NOVICE here strictly.",  
    "Advanced": "The core feedback paragraph. Apply the Instructions for Adaptation logic for ADVANCED here strictly.",  
  } }
```

[INPUT DATA]

The Quiz Problem: quiz_description_input

Student's Strategy Essay: student_strategy_input

Expert's Strategy Essay: expert_strategy_with_rubrics

Figure 11: The full system prompt used for generating post-hoc feedback

Prompt Sample: Generating Just-in-Time Feedback for Quiz Problem Solving

You are an expert Physics Education Researcher and Cognitive Scientist. Your goal is to analyze student essays describing their strategy for solving a problem. You must determine if the student's proposed strategy will lead to the Correct Answer or a specific type of Misconception.

TASK DESCRIPTION: I will provide you with a physics problem statement and a "Strategy Essay" written by a student.

1. Analyze the student's essay. Look for specific keywords or logical steps.
2. Identify if they are applying correct principles/concepts or falling into misconceptions.
3. Output your final reasoning and label as strictly valid JSON.

Label Categories & Definitions

- 'correct': The reasoning and final answer are physically sound.
- 'direction': The student makes an error where their answer is in the exact opposite direction of the correct vector OR The student does not specifically state the direction they want to apply.
- 'position': The error arises from calculating forces or properties on a different object than the one asked OR The student does not specifically state the particular object they want to calculate for.
- 'position-direction': Both a 'position' error and a 'direction' error occur simultaneously in the student's response OR The student does not specifically state the direction nor the object they want to consider.

Few-Shot Examples

[Examples of student's sample strategy essay, its error type, and feedback generated by human experts]

TEST CASE {quiz_problem}

{STUDENT_STRATEGY_ESSAY}

INSTRUCTIONS: Analyze the "Student Essay" above and provide the JSON output.

OUTPUT FORMAT:

{ "classification": Choose one from [correct, direction, position, position-direction],

"confidence": How confident the classification is (scale of 1 to 5, 5 being most confident),

"secondary_classification": What would be the second most likely label?

"feedback": Feedback message (in 50 words) to students based on their essay. If their essay is classified to make a certain type of mistake, the feedback should highlight that aspect. If it is classified as correct, the feedback can be general and notify some common mistakes that can be made. If the secondary classification is different from classification, take both classes into consideration.

}

Constraint

- Output ONLY the json.
- Do NOT provide reasoning, explanations, or introductory text.

Figure 12: The full system prompt used for generating JiT feedback

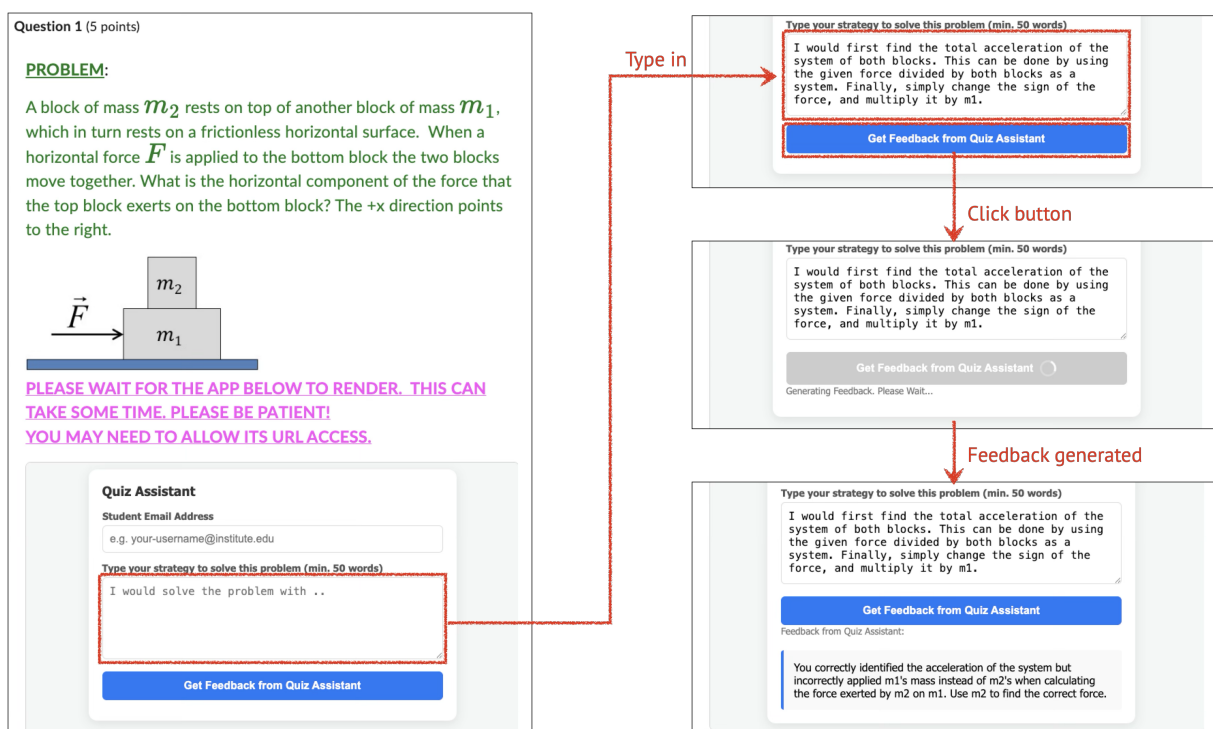


Figure 13: Overall flow of JiT feedback. The feedback snippet is placed below the quiz question. Students type in their strategy essays, click a button, and gets a feedback after a few seconds.

RABIT: Rationale-Based Distillation Towards Interpretable Automatic Speaking Assessment via a Small Language Model

Bi-Cheng Yan, Hong-Yun Lin, Fu-An Chao, Jiun-Ting Li, Berlin Chen

National Taiwan Normal University, Taipei, Taiwan

{bicheng, fuanchao, berlin}@ntnu.edu.tw

Abstract

Automatic speaking assessment (ASA) manages to quantify the language competence of foreign language learners by providing a proficiency score based on their spoken response. Existing efforts in ASA typically employ a neural grader integrated with a set of hand-crafted features to assess learners' oral proficiency from multiple facets. Despite decent performance, the black-box nature of these neural graders remains a significant barrier to providing interpretable explanations for the grading results. In light of this, we propose RABIT for ASA, a novel rationale-based knowledge distillation framework for interpretable grading decisions via a small language model. Specifically, RABIT first extracts multi-faceted grading rationales from a large language model (LLM) pertaining to the learner's response and the scoring guidelines. Subsequently, a compact yet efficient language model, equipped with distinct output heads, is jointly optimized to estimate a proficiency score while generating a sequence of grading rationales in an autoregressive manner. A series of experiments conducted on General English Proficiency Test (GEPT) dataset validates the feasibility and superiority of our method over several cutting-edge baselines.

1 Introduction

Spurred by the global demand for foreign language proficiency in both the workforce and academia, there is a growing need for the assessments of language competence (Davis and Norris, 2024). In response, the development of automatic speaking assessment (ASA) systems has garnered significant attention, figuring prominently in the fields of computer-assisted language learning (Zechner and Evanini, 2019) and large-scale language testing (Singla et al., 2021). Such systems offer a broad spectrum of applications to mitigate the disproportion between the limited number of instructors and the expanding population of foreign language learn-

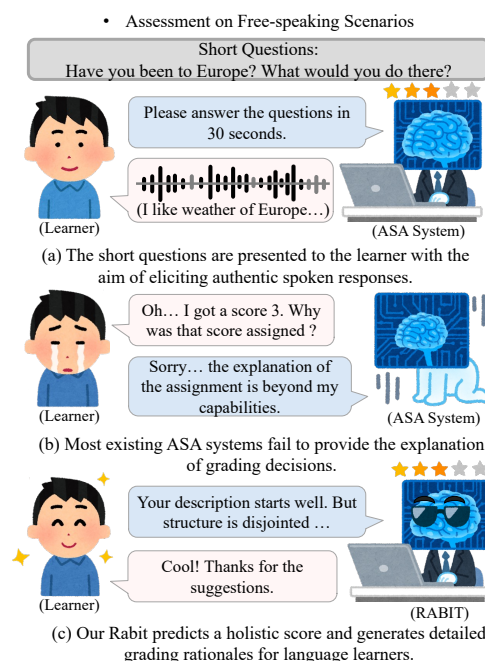


Figure 1: Motivations of our approach. (a) Assessment in the free-speaking scenario, where the ASA models aim to evaluate learner's language proficiency based on the spoken response. (b) Existing ASA systems often operate as a black-box process, lacking transparency in their grading decisions. (c) Our proposed method (RABIT) jointly optimizes proficiency estimation and rationale generation to achieve transparent and reliable assessment results.

ers, ranging from low-stakes contexts (e.g., providing informative feedback for both instructors and learners in course placement (Evanini and Wang, 2013)), to high-stakes scenarios (e.g., serving as a reliable reference for professionals in admission testing (Evanini et al., 2017; Chen and Li, 2016)).

ASA seeks to quantify the extent of language proficiency in foreign language learners across multiple dimensions, including delivery (fluency and pronunciation), language use (vocabulary and grammar), topic development (content and discourse) (Qian et al., 2019; Zechner and Evanini,

2019), and others. One of the de-facto archetypes of ASA is instantiated in free-speaking scenarios, as shown in Figure 1(a), where the language learner is presented with short, open-ended questions and instructed to respond based on their personal experiences or opinions. A leading strand of research in ASA derives a set of handcrafted features from the synergy of learners’ spoken responses and the presented short questions. These features are then fed into a neural grader to predict either a holistic score (i.e., a categorical value of overall speaking proficiency) or analytic scores (i.e., continuous numerical scores for specific aspects). Specifically, to characterize learners’ pronunciation clarity and spoken delivery, commonly used features include confidence scores of recognized linguistic units (e.g., words or phones), time-alignment information (e.g., speaking rate, pause frequency, and filled pauses), and statistical measures of fluency and pitch contours (Zechner et al., 2009; Chen et al., 2010). Grammatical accuracy and syntactic complexity are captured by the transcriptions of the learner response through part-of-speech tagging, syntactic dependency parsing, and morphological analysis (Moore et al., 2015; Qian et al., 2018). Lastly, content coherence and topical relevance are assessed by evaluating the learner response and the presented short questions, where the BERT-based semantic features combined with graph-based neural networks have been extensively investigated (Li et al., 2023; Bannò and Matassoni, 2023; Singla et al., 2021; Li et al., 2024).

Although existing efforts have achieved commendable performance on various spoken assessment tasks by leveraging a set of handcrafted features to qualify learners’ language competence, the opacity of scoring decisions remains a crucial challenge. The interpretability of scoring decisions is nearly sidelined, which might undermine the efficiency of language acquisition and diminish pedagogical utility, as depicted in Figure 1(b). In light of this, we propose a novel two-step knowledge distillation framework for ASA, termed rationale-based distillation for interpretable grading decisions (RABIT). By harnessing the reasoning capabilities of large language models (LLMs), RABIT first distills grading rationales into a lightweight yet effective small language model, which then in turn learns to generate interpretable, fine-grained scoring decisions while assessing the language proficiency of foreign language learners. To explicitly illustrate how the learner responses align with the scor-

ing rubrics, RABIT first prompts an LLM (e.g., LLaMA3.1-8B and LLaMA3.2-3B (Touvron et al., 2023)) with systematic rubric guidelines to elicit multi-faceted grading rationales conditioned on the transcriptions of learner responses and the associated short questions. Afterwards, a small language model (e.g., SmolLM2-135M-Instruct (Allal et al., 2025)) augmented with two distinct output heads is jointly optimized to predict proficiency scores and generate corresponding grading rationales in an autoregressive manner. Extensive experiments on the General English Proficiency Test (GEPT) dataset demonstrate that our methods consistently outperform several competitive baselines. Beyond performance gains, exploring a small LLM to generate grading rationales for foreign language learners introduces an additional layer of transparency to the field of ASA research.

In summary, the main contributions of this work are at least three-fold: (1) As one step towards interpretable grading decisions, RABIT is among the first attempts to evaluate language proficiency of foreign language learners while offering grading rationales, opening up a new research avenue for ASA; (2) Furthermore, to adapt small language models for ASA, we introduce prompt vectors in conjunction with a score prediction head to stabilize proficiency score estimation while maintaining the computational efficiency; (3) Extensive experiments on a realistic spoken assessment dataset, comprising authentic oral responses and expert annotations, confirm the practical and pedagogical utilities of RABIT.

2 Rationale-based Distillation for Interpretable Grading Decisions (RABIT)

This section sets out with a problem definition of ASA task and then sheds light on the proposed knowledge distillation framework, rationale-based distillation for interpretable grading decisions (RABIT). Our approach aims to improve the transparency and the practical utility of ASA through two pivotal steps, namely LLM rationale probing and small language model training, as illustrated in Figure 2(a). In the first step of RABIT, we leverage an LLM to generate multi-faceted grading rationales by conditioning on the transcriptions of learner responses, the short questions, the rubric guidelines, and the expert-annotated proficiency score. Subsequently, in the second step, these dis-

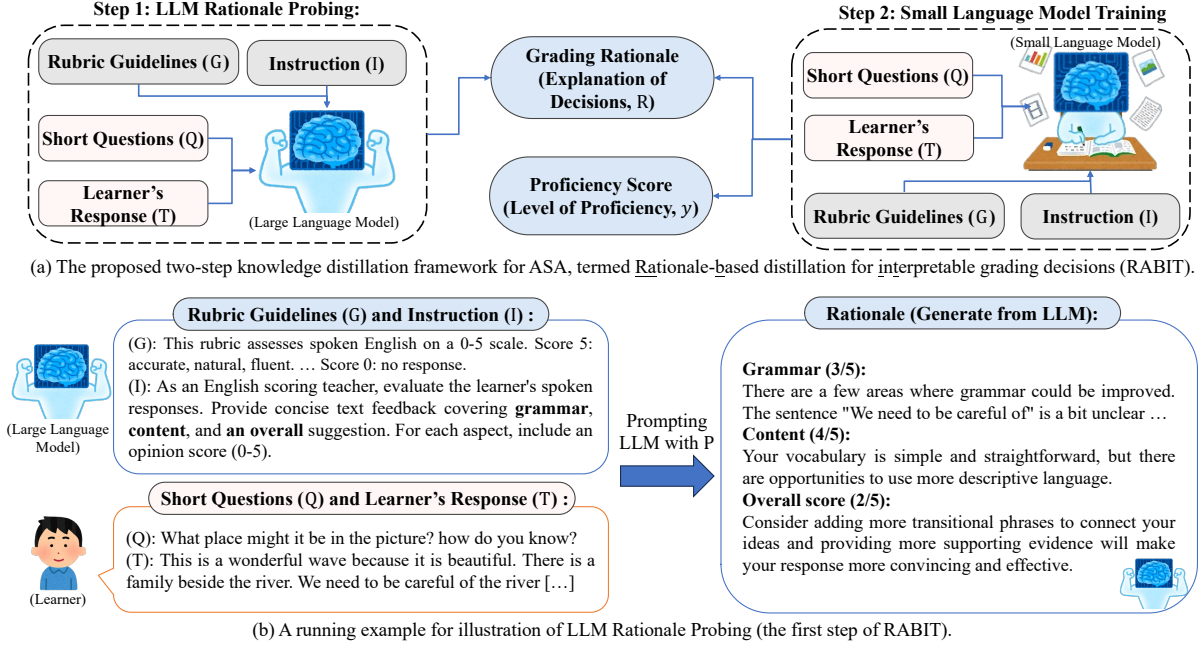


Figure 2: Overview of the proposed RABIT framework for interpretable knowledge distillation. (a) The two-stage training process, comprising LLM rationale probing and small language model (SLM) distillation. (b) A running example illustrating the rationale generation and distillation process in the first stage of RABIT.

titled rationales serve as additional supervision for a small language model, which is jointly trained to estimate proficiency scores while producing the corresponding grading rationales.

2.1 Problem Definition

Given a sequence of short questions Q , a foreign language learner is instructed to produce a spoken response X that reflects their personal experiences or opinions. An ASA system is then tasked with estimating a categorical proficiency score $y \in \{1, 2, 3, 4, 5\}$ for X , where y denotes a distinct level of language competence across multiple facets including delivery, language use, and topic development.

2.2 Step1 of RABIT: LLM Rationale Probing

For each instance of training data, consisting of a spoken response X paired with the short questions Q and the corresponding proficiency score y , we first transcribe X into a sequence of words T via a speech recognizer (e.g., Whisper-large-v3 (Radford et al., 2023)). Subsequently, a structured prompt template P is curated to elicit a sequence of grading rationales R from an LLM. The prompt template $P = [G; I; Y; Q; T]$ comprises the following components: the rubric guidelines G , which outline scoring rubrics used by human experts; the instruction I , which describes the role of the LLM in gen-

erating multi-faceted proficiency scores along with explanations; the score reference Y , which specifies y for the LLM to guide rationale elicitation; and the short questions Q coupled with the transcribed response T . The first step of RABIT is illustrated with a running example in Figure 2(b). Notably, since the rationale generation relies on the ASR transcriptions, the instruction I positions LLM as a language instructor to elicit multi-faceted grading decisions from a textual perspective, focusing on the grammatical accuracy, content relevance, and overall proficiency score. To ensure the reliability of distilled rationales, we filter out instances where the LLM-generated scores conflict with y . In preliminary experiments, we evaluated the feasibility of large multimodal language models (e.g., Phi-4-multimodal (Abouelenin et al., 2025), and QwenAudio (Chu et al., 2023)) to generate grading rationales directly from the speech signal X and the task instruction I . However, these models are prone to generating hallucinated feedback. We attribute this to the scarcity of expert-annotated ASA data and its underrepresentation during the training of these multimodal foundation models.

2.3 Step2 of RABIT: Small Language Model Training

Although LLMs have demonstrated remarkable performance on various NLP benchmarks, the high computational overhead hinders practical deployment in real-world ASA applications, particularly on resource-limited edge devices such as mobile phones or personal computers. To this end, following the extraction of rationales in Step 1, we fine-tune a small language model (e.g., SmolLM2 (Allal et al., 2025)) in Step 2 to build a computationally efficient and interpretable ASA model. As illustrated in Figure 3, we employ a small language model as the backbone, which jointly learns to predict the proficiency score and generate grading rationales based on a set of handcrafted features via parameter-efficient instruction tuning (Hu et al., 2022). Our interpretable ASA model consists of three components: proficiency-specific encoders, proficiency-specific projectors, and a small language model. In the following, we delve into the functional details of each component.

Proficiency-specific Encoders. To capture the supra-segmental pronunciation cues in the learners’ speech, we leverage a pre-trained audio encoder (e.g., Wav2vec2.0 (Baevski et al., 2020)) for delivery feature extraction (Lo et al., 2024; Chao et al., 2022; Yan et al., 2025), which takes the speech signal X as input and maps it to a sequence of high-level acoustic representations H_0^D . The language-use features H_0^L are derived from one-hot encodings of part-of-speech tags, syntactic dependency labels, and morphological features, all of which are linearly projected and packed together via individual linear transformations to form a unified representation (Peng et al., 2024). To capture the content coherence between a learner’s response and short questions, a pre-trained text encoder (e.g., BERT (Devlin et al., 2019)) is adopted to extract content features H_0^C by formatting an input text sequence as the concatenation of short questions Q and the transcribed response T , delimited by the [SEP] token to model their inter-textual relationships.

Proficiency-specific Projectors. To bridge the proficiency-specific encoders and the small language model, the projectors map the respective proficiency features into token representations, aligning the proficiency features with the latent space of the small language model. In this work, we employ a convolutional projector that encapsulates fine-grained linguistic cues into compact token rep-

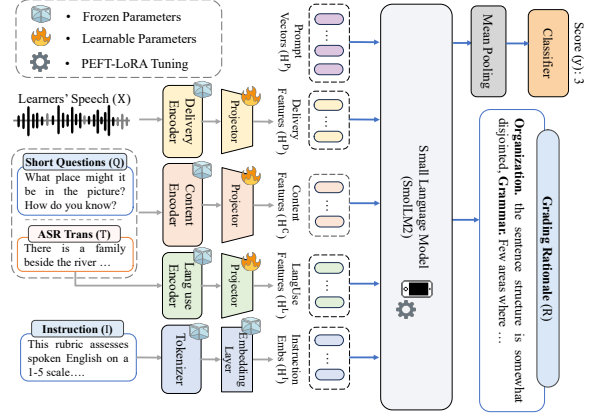


Figure 3: The proposed multi-modal ASA model in the second stage of RABIT, which jointly estimates proficiency scores and generates corresponding grading rationales.

resentations with a Macaron-style neural structure, where a Transformer block is sandwiched between two separate depth-wise convolutional layers. Each convolution layer incorporates a GELU activation function ((Lee, 2023)) and a kernel size of 3. For down-sampling, the first layer utilizes a kernel stride of $k/2$, and the second employs a kernel stride of k . We compress the delivery, language-use, and content features along the temporal dimension at reduction rates of 64, 32, and 32, respectively (i.e., $k = 16, 8$, and 8).

Small Language Model. Conditioned on a set of handcrafted features extracted from proficiency-specific encoders, our language model aims to generate a sequence of grading rationales and estimate proficiency scores for language learners. Empirically, we found that training a small language model to output both the rationale sequence and a proficiency score within a single natural language sequence often leads to inconsistent output formats. In contrast to prior works (Chu et al., 2025; Hsieh et al., 2023), our approach introduces learnable prompt vectors H^P to the inputs, which are integrated with the handcrafted features for the small language model. Following forward propagation through the language model, the outputs of these prompt vectors are pooled by an averaging operation and subsequently passed to a classification head to predict the proficiency score y . A bit more terminology: The input sequence to our language model, denoted as $H = [H^D; H^L; H^C; H^I; H^P]$, comprises prompt vectors H^P , instruction embeddings H^I and a set of handcrafted features, where H^D, H^L, H^C are the projected features derived from

the delivery, language-use, and content coherence representations (i.e., H_0^D , H_0^L , and H_0^C) via their respective projectors. H^I denotes the textual representation of the instruction I , projected through the embedding layer of the small language model. Our ASA model employs a small language model parameterized by θ to generate the grading rationales $R = (r_1, r_2, \dots, r_N)$ by minimizing the negative log-likelihood:

$$\mathcal{L}_{rat} = -\log \mathcal{P}_\theta(R|H), \quad (1)$$

$$= -\sum_{i=1}^N \log \mathcal{P}_\theta(r_i | R_{0:i-1}, H), \quad (2)$$

where $R_{0:i-1}$ denotes previously generated tokens. On a separate front, our model also predicts a proficiency score $\hat{y}$ via a classification head. To stabilize the score prediction, the classifier relies on the mean-pooled representation $\bar{h}^P$, derived by applying an average pooling operation to the output representations of prompt vectors H^P after processing through the language model. Proficiency score estimation is optimized by minimizing the cross-entropy loss between the distributions of predicted score $\hat{y}$ and the ground-truth score y (represented as a one-hot vector $\mathbf{y}$):

$$\mathcal{L}_{score} = -\sum_{c=1}^5 \mathbf{y}_c \log \hat{\mathbf{y}}_c, \quad (3)$$

$$\hat{\mathbf{y}} = \mathcal{P}_\theta(\hat{\mathbf{y}} | \bar{h}^P) = \text{softmax}(W\bar{h}^P + \mathbf{b}), \quad (4)$$

where W and $\mathbf{b}$ are learnable weights of the classification head. In the training phase, we keep the small language model frozen and integrate a LoRA module to minimize the following objective function:

$$\mathcal{L} = \mathcal{L}_{score} + \alpha \mathcal{L}_{rat}, \quad (5)$$

where $\alpha \in [0, 1]$ is a control parameter balancing the losses of rationale generation and proficiency score estimation.

3 Experimental Setups

GEPT Dataset. In this paper, a series of experiments were carried out on an in-house dataset derived from General English Proficiency Test (GEPT), a standardized assessment administered by the Language Training and Testing Center (L TTC) in Taiwan. The GEPT dataset targets English learners across various proficiency levels, comprising listening, reading, writing, and speaking.

Dataset		Human-rated Score				
		S1	S2	S3	S4	S5
Train		4	61	317	310	27
Dev		0	6	39	39	6
Test	Seen Prompt	0	11	37	39	3
	Unseen Prompt	0	4	142	137	17

Table 1: Statistics of Picture-description Section in GEPT Dataset.

For our ASA experiments, we specifically focus on the picture description section of the GEPT dataset, in which foreign language learners are presented with a picture and instructed to answer short questions. This subset consists of 1,199 recordings, each paired with an expert-annotated proficiency scores, the short questions, learner response and the corresponding pictures. Each spoken response was independently evaluated by two qualified experts adhering to the same standardized rubric. To derive a single categorical proficiency score, we averaged the two ratings and applied the floor function. The GEPT dataset was partitioned into training, development, and seen-prompt test sets in an 8:1:1 ratio. An isolated unseen-prompt test set comprising 300 recordings was reserved to evaluate the generalization capability of the proposed model to novel picture description learning scenario. The dataset statistics are provided in Table 1.

Implementation Detailed. In the first step of RA-BIT, we employed LLaMA-3.1-8B-Instruct for rationale generation. In the second step, we used SmolLM2 model family as the backbone for our interpretable ASA model. To extract various hand-crafted features, we adopted Wav2vec2.0¹ to extract delivery features. Content coherence was modeled using a BERT-based text encoder², while language-use features were derived from word-level linguistic units. Following Peng et al. (2024), language-use features include POS tags, dependency labels, and morphological features processed by the spaCy toolkit³ based on the transcription of learner responses. As for the model training, we employed the Adam optimizer with a learning rate

¹<https://huggingface.co/facebook/wav2vec2-large-xlsr-53>

²<https://huggingface.co/google-bert/bert-base-uncased>

³<https://spacy.io/>

Model	Seen-Prompt Test			Unseen-Prompt Test		
	Accuracy	Weighted-F1	Macro-F1	Accuracy	Weighted-F1	Macro-F1
⁺ Qian2019	55.56	54.80	–	64.33	62.80	–
⁺ SAMAD	65.56	64.80	–	69.67	68.40	–
Content Model	60.44 (± 3.20)	59.41 (± 3.30)	48.28 (± 3.85)	67.00 (± 1.39)	65.14 (± 0.99)	38.56 (± 3.20)
Delivery Model	52.00 (± 2.88)	48.41 (± 4.70)	31.72 (± 4.85)	52.33 (± 2.68)	49.42 (± 2.81)	26.89 (± 1.35)
Multi-ASA	63.56 (± 2.88)	62.85 (± 2.92)	44.60 (± 5.15)	67.00 (± 5.15)	65.37 (± 4.65)	41.24 (± 4.49)
Multi-ASA-Rat	66.89 (± 1.64)	66.31 (± 1.90)	45.07 (± 3.70)	68.40 (± 1.83)	66.67 (± 2.31)	38.67 (± 3.44)
RABIT	70.00 (± 1.11)	69.09 (± 1.10)	56.54 (± 6.97)	71.66 (± 1.52)	69.71 (± 1.70)	47.39 (± 1.47)

Table 2: Performance comparison with state-of-the-art methods on the GEPT dataset in terms of Accuracy (%), Weighted F1-score (%), and Macro F1-score (%). Standard deviations over 3 runs are reported in parentheses.⁺Results for Qian2019 and SAMAD are reported in Peng et al. (2024).

of 0.001 and a batch size of 3. A warm-up strategy was applied, initiated with 1/100 of the maximum learning rate, and adjusted over steps with a cosine scheduler. All experiments were implemented with PyTorch and executed on a single NVIDIA 2080Ti GPU. To ensure reproducibility and mitigate the effects of randomness, we conducted three independent trials with different random seeds. Each model was trained for 100 epochs, and performance is reported as the average across the three runs. The optimal checkpoint for each trial was selected based on the minimum validation cross-entropy loss. We are committed to making all code, configurations, and logs available upon acceptance.

Evaluation Metric. We evaluate the performance of our model using the following metrics: 1) Classification accuracy (%), representing the percentage of correctly predicted proficiency levels ranging from 1 to 5; 2) Macro F1-score (%), which averages the F1-scores across all proficiency levels, treating each class with equal importance; and 3) Weighted F1-score (%), which weights the F1-score of each proficiency score by its support (number of true instances) to account for label imbalance. The weighted F1-score is particularly crucial for multi-class classification tasks involving imbalanced or skewed labels, as it offers a more representative evaluation of overall performance.

Comparative Methods. To evaluate the performance of our proposed framework, we compare RABIT with the following state-of-the-art (SOTA) methods. 1) Conventional methods: **Qian2019** is an iconic ASA model that extracts various handcrafted features from transcriptions and the cor-

responding time-alignment information derived from learner speech (Qian et al., 2019). Subsequently, **SAMAD** advances the neural architecture of Qian2019 by incorporating Transformer blocks to capture long-term dependencies of linguistic cues, while introducing soft-label optimization to mitigate the label imbalance problem in ASA model training (Peng et al., 2024). 2) Multi-faceted neural graders: **Multi-ASA** employs the same proficiency-specific encoders as RABIT to extract delivery, language-use, and content-related features. In contrast to the use of small language model for proficiency assessment, Multi-ASA adopts a simple concatenation for feature fusion. The fused features are subsequently passed through a linear projection layer followed by a mean pooling operation to predict the proficiency score. Moreover, to assess the efficacy of grading rationales in ASA, **Multi-ASA-Rat** integrates grading rationales as auxiliary input. We leverage a BERT encoder to extract semantic representations from the grading rationales, which are then fused with handcrafted features within the Multi-ASA to collaborative proficiency enhance proficiency score estimation. 3) Single-faceted neural graders: To analyze the contribution of each modality-specific encoder in RABIT, we report the assessment performance using only the content encoder (**Content Model**) and delivery encoder (**Delivery Model**). Both models extract the corresponding features and assess the proficiency score via simple mean pooling followed by a linear classifier.

4 Experimental Results

Main Results and Performance Benchmarking.

At the outset, we compare the proposed approach (viz. RABIT) with existing state-of-the-art ASA methods. From the results shown in the Table 2, we have the following observations. 1) The proposed RABIT consistently outperforms all baseline methods on both seen- and unseen-prompt test sets in terms of accuracy and F1-scores, demonstrating that RABIT surpasses conventional baselines as well as single- and multi-faceted neural graders, while highlighting the strong generalization capability of our framework across diverse evaluation scenarios. 2) Regarding multi-faceted neural graders, Multi-ASA-Rat achieves notable performance gains over its base model (Multi-ASA). By incorporating LLM-generated grading rationales as auxiliary inputs, the results demonstrate that semantic encodings of rationales are highly complementary to other handcrafted features, leading to enhanced assessment accuracy in ASA. In contrast, RABIT formulates rationale generation as a primary learning objective and predicts language proficiency through a small language model, which not only achieves superior performance compared to Multi-ASA-Rat but also significantly enhances the transparency of the assessment results. Furthermore, in a comparison within the single-faceted neural graders, it is evident that the Content Model achieves better results than the Delivery Model. This finding suggests that content coherence exerts a greater influence than spoken delivery in ASA, aligning with observations reported in previous studies (Qian et al., 2019; Bannò and Matassoni, 2023). Moving beyond individual model components, Multi-ASA integrates handcrafted features from content and delivery, alongside language-use features, into a unified neural architecture, resulting in superior performance over its single-faceted counterparts (viz. Content and Delivery models). 3) As to the conventional methods, SAMAD exhibits strong performance on both test sets, significantly outperforming Qian2019 and achieving competitive results to Multi-ASA-Rat. Moreover, SAMAD achieves superior results on the test set with unseen-prompts compared to other baseline methods. These performance gains primarily stem from its delivery modeling, which leverages ASR transcriptions coupled with the spoken signals to capture prosody and fluency of learner speech, effectively boosting the generalization capability

Setting	Seen Prompt		Unseen Prompt	
	Acc	wF1	Acc	wF1
Number of Prompts				
1	69.63	68.87	70.66	69.06
3*	70.00	69.09	71.66	69.71
5	68.51	67.64	71.44	69.49
Combination Weight for Rationale Generation				
(In a 3-prompt vector setting)				
0.0	66.67	66.18	70.33	68.85
0.3	69.63	68.71	71.22	69.28
0.5*	70.00	69.09	71.66	69.71
0.7	69.63	68.71	70.89	68.70
Impact of Base LM Choice (SmolLM2)				
(In a 3-prompt vector and weight setting of 0.5)				
135M-Instruct*	70.00	69.09	71.66	69.71
360M-Instruct	68.15	67.45	71.56	69.47

Table 3: Ablations of parameters in RABIT, reporting accuracy (Acc, %) and weighted F1-score (wF1, %).

*Denotes the setting reported in Table 2.

across evaluation scenarios.

Ablation Study on RABIT Parameters. In this section, we perform an ablation study to systematically examine the assessment performance of RABIT under various parameter configurations on both seen- and unseen-prompt test sets. We first investigate the impact of the number of prompt vectors (i.e., $|\mathbf{H}^P| \in \{1, 3, 5\}$) employed in RABIT. The corresponding results are summarized in the first part of Table 3. Notably, RABIT facilitates proficiency score prediction through the integration of prompt vectors into the model input, rather than relying solely on natural language generation. As the number of prompt vectors increases, we observe performance gains on the unseen-prompt test set; however, these improvements remain marginal. Conversely, a noticeable decline in performance is evident on the seen-prompt test set. To strike an optimal balance between generalization across test sets and computational efficiency, we configure the RABIT with $|\mathbf{H}^P| = 3$.

Subsequently, we investigate the combination weight α that balances the two learning objectives (i.e., proficiency estimation $\mathcal{L}_{score}$ and grading rationale generation $\mathcal{L}_{rat}$), as defined in Eq. (5). The weight parameter α is selected from

$\{0.3, 0.5, 0.7\}$. Furthermore, $\alpha = 0.0$ is included as a reference baseline variant, representing the performance without rationale guidance. As demonstrated in the second part of Table 3, our results suggest that the rationale generation task is crucial for the performance of ASA; when this task is omitted (i.e., $\alpha = 0.0$), the corresponding performance is significantly inferior to all other settings with rationale guidance ($\alpha > 0.0$). Furthermore, we find that the choice of combination weight has a relatively modest impact on overall performance. Increasing α from 0.3 to 0.7 produces only marginal gains on the seen-prompt test set, while inducing a slight performance degradation in unseen-prompt test set.

Finally, the third part of Table 3 examines how the scale of the backbone language model impacts ASA performance. Notably, our results demonstrate that RABIT, when configured with SmolLM-135M-Instruct, consistently outperforms its larger 360M counterpart across all evaluation metrics. This finding suggests that for RABIT, a compact model offers a more balanced model capacity, which is sufficient to capture grading nuances on seen prompts while effectively generalizing to the unseen-prompt test set. Considering both predictive power and computational efficiency, RABIT adopts SmolLM2-135M-Instruct as its default backbone model.

Qualitative Examination of Grading Rationales.

Building on the evidence that rationale generation enhances the ASA performance of RABIT, we further conduct a qualitative comparison to evaluate the interpretability of the grading rationales generated by RABIT against those from the teacher LLM (viz. the source LLM used for distillation). As shown in Figure 4, the majority of the grading rationales produced by RABIT and the source LLM share similar semantic content (sentences highlighted in green). However, as illustrated by the sentences marked in purple, RABIT provides grammatical feedback by identifying errors rather than generating full corrected sentences. Notably, RABIT retains the emergent capabilities inherited from its base model, SmolLM2, allowing it to offer nuanced grading opinions or instructive feedback on the spoken response (sentences highlighted in orange). Moreover, as the rationale generation process is conditioned on the expert-annotated proficiency score, the generated rationales and proficiency scores exhibit a high degree of consistency. These qualitative results confirm that RABIT not only effectively captures the grading rationales of

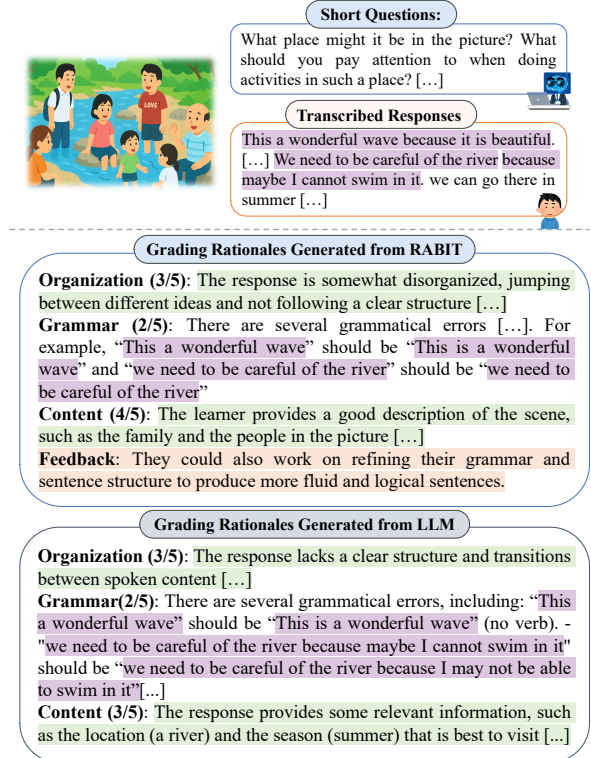


Figure 4: Qualitative comparisons of rationales generated by RABIT and the source LLM. Sentences highlighted in green indicate semantic consistency with the rationales generated by the LLM, while those in purple reflect semantic divergence. Sentences in orange highlight emergent capabilities exhibited by RABIT.

the teacher model but also preserves the intrinsic reasoning capabilities of the underlying small language model.

5 Conclusion and Future Work

In this paper, we presented an ASA model designed to provide a transparent grading process without compromising computational efficiency. This advancement highly broadens the scope of ASA system in the real-world applications. As a pivotal step toward interpretable assessment, we proposed RABIT, a rationale-based distillation framework that produces grading decisions while simultaneously estimating the proficiency scores of language learners via a small language model. To enhance the stability of proficiency score estimation, we introduced prompt vectors for the small language model, making a pioneering effort within the field of ASA. Extensive experiments on the picture-description section of the GEPT dataset validate the superiority of RABIT, revealing that integrating rationale generation during the learning process further elevates ASA performance.

Limitations

The proposed method is limited by the quality of the rationales generated from LLMs. Additionally, the visual information contained in the prompts (i.e., pictures) is nearly sidelined in current work. Despite this minimal integration of visual data, RABIT remains consistent with the standard ASA rubrics of the GEPT. These scoring rubrics prioritize spoken delivery and language use, while treating topic development as only a secondary facet of the overall proficiency score. To address these limitations, future work will investigate integrating visual features from the picture prompts and improving the reliability of rationale generation through multi-agent prompting or human feedback mechanisms. Furthermore, we plan to enhance the diversity of the generated rationales of RABIT by leveraging back-translation techniques (Shen et al., 2025).

Ethics Statement

This study adheres to the ACL Code of Ethics. We utilized anonymized datasets provided by LTTC under authorized access, ensuring that data curation aligns with ACL standards. While RABIT provides automated proficiency assessments, it is designed to supplement human raters and provide informative feedback for language learners and instructors in low-stakes learning scenarios. We also acknowledge potential biases inherent in teacher LLMs and encourage cautious interpretation of automated feedback.

References

- Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.
- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, and 1 others. 2025. Smollm2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems (NIPS)*, 33:12449–12460.
- Stefano Bannò and Marco Matassoni. 2023. Proficiency assessment of 12 spoken english using wav2vec 2.0. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1088–1095. IEEE.
- Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2022. 3m: An effective multi-view, multi-granularity, and multi-aspect modeling approach to english pronunciation assessment. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 575–582. IEEE.
- Lei Chen, Keelan Evanini, and Xie Sun. 2010. Assessment of non-native speech using vowel space characteristics. In *IEEE Spoken Language Technology Workshop*, pages 139–144. IEEE.
- Nancy F Chen and Haizhou Li. 2016. Computer-assisted pronunciation training: From pronunciation scoring towards spoken language learning. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–7. IEEE.
- SeongYeub Chu, Jong Woo Kim, Bryan Wong, and Mun Yong Yi. 2025. Rationale behind essay scores: Enhancing s-llm’s multi-trait essay scoring with rationale generated by llms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5796–5814.
- Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.
- Larry Davis and John M Norris. 2024. *Challenges and innovations in speaking assessment*. Routledge.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL)*, pages 4171–4186.
- Keelan Evanini, Maurice Cogan Hauck, and Kenji Hakuta. 2017. Approaches to automated scoring of speaking for k–12 english language proficiency assessments. *ETS Research Report Series*, 2017(1):1–11.
- Keelan Evanini and Xinhao Wang. 2013. Automated speech scoring for non-native middle school students with multiple task types. In *Proceedings of the Inter-speech (INTERSPEECH)*, pages 2435–2439.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay

- Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.
- Minhyeok Lee. 2023. Gelu activation function in deep learning: a comprehensive mathematical analysis and performance. *arXiv preprint arXiv:2305.12073*.
- Jiun-Ting Li, Tien-Hong Lo, Bi-Cheng Yan, Yung-Chang Hsu, and Berlin Chen. 2023. Graph-enhanced transformer architecture with novel use of ceft vocabulary profile and filled pauses in automated speaking assessment. In *SLaTE*, pages 109–113.
- Jiun-Ting Li, Bi-Cheng Yan, Tien-Hong Lo, Yi-Cheng Wang, Yung-Chang Hsu, and Berlin Chen. 2024. Automated speaking assessment of conversation tests with novel graph-based modeling on spoken response coherence. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 825–832. IEEE.
- Tien-Hong Lo, Fu-An Chao, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2024. An effective automated speaking assessment approach to mitigating data scarcity and imbalanced distribution. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1352–1362.
- Russell Moore, Andrew Caines, Calbert Graham, and Paula Buttery. 2015. Incremental dependency parsing and disfluency detection in spoken learner english. In *International Conference on Text, Speech, and Dialogue (TSD)*, pages 470–479. Springer.
- Wen-Hsuan Peng, Sally Chen, and Berlin Chen. 2024. Enhancing automatic speech assessment leveraging heterogeneous features and soft labels for ordinal classification. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 945–952. IEEE.
- Yao Qian, Patrick Lange, Keelan Evanini, Robert Pugh, Rutuja Ubale, Matthew Mulholland, and Xinhao Wang. 2019. Neural approaches to automated speech scoring of monologue and dialogue responses. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8112–8116. IEEE.
- Yao Qian, Rutuja Ubale, Matthew Mulholland, Keelan Evanini, and Xinhao Wang. 2018. A prompt-aware neural network approach to content-based scoring of non-native spontaneous speech. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 979–986. IEEE.
- 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)*, pages 28492–28518. PMLR.
- Zhanming Shen, Hao Chen, Yulei Tang, Shaolin Zhu, Wentao Ye, Xiaomeng Hu, Haobo Wang, Gang Chen, and Junbo Zhao. 2025. Cycle-instruct: Fully seed-free instruction tuning via dual self-training and cycle consistency. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5123–5137.
- Yaman Kumar Singla, Avyakt Gupta, Shaurya Bagga, Changyou Chen, Balaji Krishnamurthy, and Rajiv Ratn Shah. 2021. Speaker-conditioned hierarchical modeling for automated speech scoring. In *Proceedings of the ACM international conference on information & knowledge management (CIKM)*, pages 1681–1691.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Bi-Cheng Yan, Ming-Kang Tsai, and Berlin Chen. 2025. Muffin: Multifaceted pronunciation feedback model with interactive hierarchical neural modeling. *IEEE Transactions on Audio, Speech and Language Processing*.
- Klaus Zechner and Keelan Evanini. 2019. *Automated speaking assessment: Using language technologies to score spontaneous speech*. Routledge.
- Klaus Zechner, Derrick Higgins, Xiaoming Xi, and David M Williamson. 2009. Automatic scoring of non-native spontaneous speech in tests of spoken english. *Speech communication*, 51(10):883–895.

Towards Pedagogically Aligned LLM Tutors for Math Mistake Remediation

Kseniia Petukhova¹, Tien Dat Nguyen¹, Ekaterina Kochmar^{1,2}

¹Mohamed bin Zayed University of Artificial Intelligence, ²University of Victoria
{kseniia.petukhova, tiendat.n, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

Large language models have strong potential for use in intelligent tutoring systems, but they often fail to follow effective pedagogical strategies, such as guiding students without revealing final answers. We study the application of a two-stage alignment pipeline for math mistake remediation, combining supervised fine-tuning on tutoring dialogs with Direct Preference Optimization on synthetic preference pairs. We construct a dataset that integrates existing tutoring corpora with synthetic data generated along pedagogical dimensions, such as scaffolding and factuality, and study different input configurations that incorporate solution correctness and gold answers. Experiments show that this approach improves both factual accuracy and pedagogical quality over base models and existing tutoring models. Human evaluation further indicates that our best model is competitive with a strong proprietary baseline, while providing additional benefits in terms of openness, transparency, and reproducibility. Our results highlight the effectiveness of preference-based pedagogical alignment, while also revealing challenges in reliably evaluating tutoring quality. The code and data are released at <https://github.com/Kpetyxova/towards-aligned-math-tutor>.

1 Introduction

Human tutoring plays a fundamental role in education, supporting learner development and contributing to broader societal advancement. One-on-one tutoring has long been shown to be highly effective (Bloom, 1984), yet the scarcity of qualified human tutors limits its large-scale adoption. Recent progress in large language models (LLMs) has opened new opportunities for educational applications (Wang et al., 2024; Gan et al., 2023), enabling the development of LLM-driven intelligent tutoring systems (ITSs) (Pal Chowdhury et al., 2024; Liu et al., 2024b) as well as tutoring through prompting approaches (Jurenka et al., 2024; McNichols

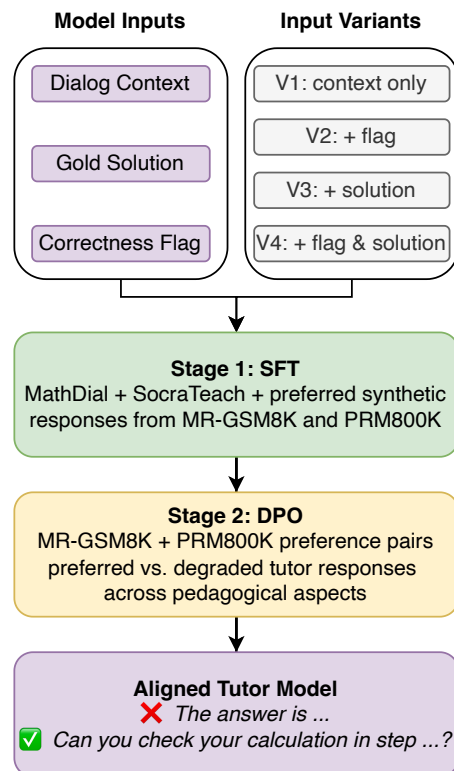


Figure 1: Overview of our two-stage pedagogical alignment pipeline. We first perform SFT on tutoring dialogs, then apply DPO using synthetic preference pairs for mistake remediation. We compare four input configurations (V1–V4) that vary in access to the student answer correctness flag and the gold solution.

et al., 2023; Mollick and Mollick, 2024). These AI-based tutors support a wide range of instructional goals (Wollny et al., 2021), with particular emphasis on addressing student errors and misunderstandings – an area that continues to drive research in AI tutoring (Macina et al., 2023; Wang et al., 2023).

Although LLMs are capable of producing natural-sounding conversations and performing well on various reasoning tasks, including commonsense and elementary mathematical reason-

ing (Achiam et al., 2023; Kojima et al., 2022; Laskar et al., 2023; Yang et al., 2024), they are not immediately suitable for deployment in educational settings without careful adaptation. High-quality tutoring involves more than generating fluent responses – it requires guiding learners toward constructing their own understanding. Effective tutors rely on pedagogical strategies such as providing hints, engaging students through Socratic questioning (Carey and Mullan, 2004), and promoting active problem-solving. Consequently, LLM-based tutoring systems should be designed to align with established human tutoring practices (Nye et al., 2014) and active learning principles that have been shown to improve learning outcomes (Freeman et al., 2014).

To build reliable and pedagogically sound tutoring systems, models must embody these instructional principles. In particular, they should be factually accurate, avoid prematurely revealing final answers, and instead guide students toward understanding and self-correction. Existing tutoring models attempt to incorporate these principles to varying degrees. For example, SocraticLM (Liu et al., 2024a) adopts a Socratic “thought-provoking” paradigm, guiding students with step-by-step questions that encourage active participation and independent reasoning rather than directly providing answers. In contrast, TutorRL-7B (Dinucu-Jianu et al., 2025) optimizes tutoring behavior through reinforcement learning over simulated student-tutor interactions, using reward functions that balance pedagogical quality and student success (solve rate). While these approaches demonstrate promising results, they also highlight the remaining challenges. SocraticLM relies on the supervised imitation of pedagogical behavior, which may limit control over specific pedagogical attributes. Similarly, TutorRL-7B optimizes tutoring as a holistic policy using scalar rewards, where pedagogical behaviors are learned implicitly through interaction and reward design. This makes it difficult to explicitly control or disentangle individual aspects of tutoring, such as factual accuracy, mistake identification, or the degree of solution revealing. We argue that explicitly modeling and controlling these aspects can lead to more robust and interpretable tutoring systems. Motivated by this, we propose a two-stage pedagogical alignment pipeline (Figure 1) that combines supervised fine-tuning (SFT) with direct preference optimization (DPO) to align tutor responses with desired pedagogical behaviors

directly. We further study different input configurations, varying the information provided to the model (e.g., correctness flags and gold solutions) to better disentangle sub-tasks such as error detection and response generation.

In summary, our key contributions are as follows:

- We construct a dataset for math mistake remediation by combining existing tutoring datasets with synthetic data derived from math-task datasets containing LLM-generated solutions. We treat these solutions as student responses and generate corresponding tutor responses using GPT-5, along with degraded variants across multiple pedagogical dimensions, resulting in a set of preference pairs for DPO.
- We study different input configurations, analyzing how access to additional information (e.g., student solution correctness flags and gold solutions) affects model performance.
- We fine-tune both open-source and proprietary models using SFT and DPO on the constructed dataset, showing that our best open-source-aligned model outperforms a strong proprietary baseline in the human evaluation.
- We release all resources developed in this work, including the synthetic dataset, trained models, and code, to support further research on pedagogically aligned AI tutors: <https://github.com/Kpetyxova/towards-aligned-math-tutor>.

2 Related work

2.1 Pedagogical alignment methods for LLM-based tutors

Large language models are frequently optimized for general-purpose helpfulness, which often leads to tutoring behaviors that are suboptimal for learning, such as prematurely revealing solutions or failing to diagnose student misconceptions. We refer to the objective of shaping the model’s behavior toward guided problem-solving, targeted feedback, and adaptive scaffolding as *pedagogical alignment*. Early approaches address this by using tutoring-specific SFT, in which models are trained to imitate the pedagogical dialog style. SocraticLM (Liu et al., 2024a), for instance, fine-tunes ChatGLM3-6B (GLM et al., 2024) on large-scale Socratic dialogs

generated via a Dean-Teacher-Student multi-agent pipeline, yielding consistent gains across multiple teaching skill dimensions. However, its training remains a supervised imitation of synthesized teacher responses, rather than an explicit optimization of a preference-based objective over pedagogical attributes, which can limit direct control over trade-offs (e.g., hint granularity vs. answer revealing) and robustness to unseen tutoring situations.

In parallel, LearnLM (Team et al., 2024) introduces *pedagogical instruction following*, applying reinforcement learning from human feedback (RLHF) with human preference data to improve adherence to system-level pedagogical instructions in multi-turn settings. Although effective, this approach depends on large-scale human preference collection and expert evaluation infrastructure.

More recently, online reinforcement learning has been explored over simulated student-tutor interactions to optimize multi-turn teaching behavior. TutorRL (Dinucu-Jianu et al., 2025), for instance, trains tutoring policies via on-policy reinforcement learning using simulated student dialogs, optimizing rewards that reflect both student learning outcomes (student solve rate) and pedagogical quality assessed by LLM-based judges.

A more scalable alternative is offline preference optimization with synthetic supervision. Sonkar et al. (2024) show that DPO (Rafailov et al., 2023) and related objectives (KTO (Ethayarajh et al., 2024), IPO (Azar et al., 2023)) consistently outperform SFT in shifting model behavior toward scaffolded guidance, with gains that remain stable across longer multi-turn dialogs. The effectiveness of such methods depends critically on preference data quality: RMBoost (Shen et al., 2024) addresses this by pre-selecting a preference label and conditionally generating a contrastive response guided by predefined multi-aspect evaluation criteria, reducing label noise in synthetic pair construction. Our work follows the offline preference optimization paradigm, applying preference-conditional generation to pedagogical dimensions for remediation of math mistakes.

2.2 Datasets for pedagogical alignment and mistake remediation

Training pedagogically aligned tutoring models requires data that capture both tutoring interaction structure and student error patterns. On the dialog side, SocraTeach (Liu et al., 2024a) provides large-scale synthetic Socratic tutoring di-

alogs with various student cognitive profiles, while MathDial (Macina et al., 2023) is collected by pairing human teachers with LLM-simulated students exhibiting common math errors, annotated with a taxonomy of teacher moves that captures realistic scaffolding strategies. In our work, both datasets serve as instruction tuning data for the SFT stage, where each tutor turn is treated as a supervised target conditioned on the prior dialog context. Beyond dialog structure, grounding of mistake remediation scenarios also requires fine-grained signals about student errors. MR-GSM8K (Zeng et al., 2023) provides structured error-centric instances by defining the evaluation as meta-reasoning over solution correctness, while PRM800K (Lightman et al., 2023) contributes step-level correctness labels on MATH (Hendrycks et al., 2021) solutions, enabling precise identification of where reasoning errors occur. We repurpose both datasets to construct preference pairs, where tutor responses are generated to contrast pedagogically effective responses with suboptimal ones.

2.3 Evaluation methods for pedagogical ability

Evaluating tutoring quality is inherently multi-dimensional: an effective tutor response must identify the student’s mistake, locate where it occurs, provide guidance without revealing the answer, and offer actionable next steps. Previous work, such as Liu et al. (2024a) and Team et al. (2024), relies on human ratings or scenario-based expert rubrics, which are reliable but expensive and difficult to scale. This has motivated the development of automated taxonomy-based evaluation frameworks grounded in learning sciences.

Maurya et al. (2025) propose a unified eight-dimension taxonomy for mistake remediation and release MRBench; they further directly assess LLM-based judging and find that Prometheus2’s (Kim et al., 2024) pedagogical annotations correlate poorly (often negatively) with human labels on MRBench, suggesting that LLM-as-judge evaluation is unreliable for fine-grained tutoring dimensions. In parallel, the BEA 2025 Shared Task (Kochmar et al., 2025) formulates four pedagogical dimensions as supervised prediction problems and shows that scalable automatic evaluation remains challenging (best 3-class macro F1 ≈ 0.58 – 0.72), despite substantial inter-annotator agreement (Fleiss’ $\kappa \approx 0.65$).

AITutor-EvalKit (Naeem et al., 2025) ad-

dresses this by providing an end-to-end evaluation toolkit for student mistake remediation (SMR), featuring a lightweight multi-task model (LoMTL) trained to assess the quality of tutor responses along these four dimensions jointly. We adopt this evaluation framework in our work because it aligns closely with our task definition and provides scalable, aspect-level feedback matching our training objectives.

3 Experimental Setup

3.1 Data

We use the MathDial and SocraTeach datasets to fine-tune our models on dialog data. Since both datasets consist of tutor-student conversations (2,861 in MathDial and 35K in SocraTeach), we preprocess each conversation by splitting it into dialog turns. Each tutor turn is treated as a training instance, where all preceding turns serve as context, and the current tutor utterance is used as the gold response. For each instance, we have the gold solution to the underlying math problem and a flag indicating whether the student has solved the task correctly. In MathDial, we assign the flag *No* to all student turns except the final one; for the final turn, we assign *Yes* if the self-correctness flag provided in the dataset is *True*. In SocraTeach, we assign *No* to all student turns except the final one, as dialogs are constructed to terminate after the student provides a correct solution and the tutor acknowledges it. This preprocessing yields 18,609 instances from MathDial and 171,296 from SocraTeach.

To align the models with pedagogical values and to augment the data beyond MathDial and SocraTeach, we adopt the RMBoost approach (Shen et al., 2024) to generate synthetic data from the MR-GSM8K and PRM800K datasets. Since PRM800K includes mathematical problems spanning a wide range of difficulty levels, with metadata indicating topic and difficulty (where levels correspond to relative difficulty tiers of competition-style math problems rather than curriculum-based grade levels), we filter it to leave only algebra problems of levels 1-2, number theory problems of level 1, and pre-algebra problems of levels 1-3. We convert each instance in these datasets into a single dialog turn: the tutor first introduces the task, and the student responds with a solution. The student response corresponds to the LLM-generated step-by-step solution provided in the original datasets. When the solution is incorrect, we use GPT-5 to generate an

appropriate tutor response. Conditioning on this response, we then use GPT-4.1 to generate a lower-quality version by degrading specific aspects of the response. We use GPT-5 to generate high-quality tutor responses, motivated by its stronger capabilities compared to GPT-4.1; this assumption was validated through manual inspection of generated responses. We use GPT-4.1 to produce degraded variants, as it does not require a model of comparable capacity. We adapt the prompt structure from RMBoost, modifying the evaluation aspects to reflect pedagogical criteria. Specifically, we define the following pedagogical criteria: (1) *Factuality* – the response should be factually correct, should not contradict what the student has said, and should not contain irrelevant information; (2) *Mistake Identification* – the response should identify, either explicitly or implicitly, that there is a mistake in the student’s solution (for example, saying “Nice try” would miss this aspect); (3) *Targetedness* – the response should address the core misconception/misunderstanding of a student; (4) *Revealing Answer* – while it is sometimes necessary and acceptable to share the answer to a substep, the tutor should avoid giving away the final answer; (5) *Clarity* – the tutor’s response should be free of awkward, confusing or misleading wording. The prompts used in this process are shown in Appendix A. This results in a dataset of 29,390 preference pairs.

Because MR-GSM8K and PRM800K also include instances with correct student solutions, we use these cases to create dialog examples without student errors, ensuring that the model does not always assume that the student’s solution is incorrect. For these instances, we prompt GPT-5 to generate tutor responses appropriate for correct solutions and randomly assign them as follow-up turns. This process yields 3,769 additional instances. The prompt used for this procedure is shown in the Appendix B.

The statistics of the collected dataset are shown in Table 1.

3.2 Model Training

We train our models using a two-stage pipeline consisting of SFT followed by preference optimization via DPO (Rafailov et al., 2023). All stages use parameter-efficient fine-tuning via LoRA (Hu et al., 2021).

We first fine-tune the base model on tutor feedback data using standard cross-entropy loss on the tutor responses. Training runs for up to 5 epochs using the AdamW and a cosine learning rate sched-

Dataset	SFT Turns		DPO Pairs
	Correct	Incorrect	
MathDial	2,855	15,754	–
SocraTeach	46,104	125,192	–
MR-GSM8K	1,390	1,387	6,935
PRM800K	2,379	4,491	22,455
Total	52,728	146,824	29,390

Table 1: Overview of the dataset statistics. Synthetically generated pairs are used for both DPO training and SFT (on preferred responses).

ule. We apply early stopping based on validation loss to prevent overfitting. The resulting LoRA is used to initialize the next training stage.

In the second stage, we align the model with pedagogical preferences using DPO on the synthetic preference dataset.

Manual inspection of our preference data revealed that responses with degraded *Clarity* are not consistently worse; in many cases, they are simply paraphrases that preserve similar pedagogical value (see Appendix F). Therefore, we do not treat this aspect as equally important as the others. Specifically, we weight each training sample by a factor w_i reflecting the importance of its associated pedagogical aspect: *Factuality*, *Mistake Identification*, *Targetedness*, and *Revealing Answer* are each assigned a weight of 1, while *Clarity* is assigned a weight of 0.5. Full details of the weighted DPO objective are in Appendix G.

We use Qwen3-4B-Instruct-2507 and Qwen3-8B (Yang et al., 2025) as our pre-trained base models. In addition, we include GPT-4.1-nano,¹ a proprietary model from a different family, to provide cross-family comparison. For GPT-4.1-nano, fine-tuning is performed using the SFT and DPO functionality available through the provider’s interface. As a result, the training procedure is not fully identical to that used for the Qwen models.

3.3 Experiments

We compare four configurations:

- V1: The model conditions only on the dialog context when generating the next tutor response;
- V2: The model is additionally provided with a binary flag indicating whether the student’s

solution is correct;

- V3: The model is additionally provided with the gold solution to the underlying math problem;
- V4: The model is additionally provided with both the correctness flag and the gold solution.

This setup allows us to isolate the effect of different levels of available information on response quality. In particular, configurations that provide the correctness flag and/or the gold solution (V2-V4) assume access to signals that may not always be directly available in real-world deployment. However, such settings are consistent with the standard human tutoring practice, where tutors have access to the correct solutions when providing feedback. In practice, such information would need to be obtained either from curated solution sources or by automatically solving the task. The latter introduces an additional source of error, as mistakes in the solution process may propagate to the generated feedback. Our current work does not model this setting explicitly. Therefore, configurations with access to gold solutions can be interpreted as upper bounds on performance, reflecting a scenario with reliable solution supervision. The prompt template used for training is provided in Appendix C.

3.4 Baselines

We compare our models against the following baselines:

- Base models: Qwen3-4B-Instruct-2507, Qwen3-8B, and GPT-4.1-nano.
- GPT-5: An advanced proprietary model.
- SocraticLM: ChatGLM3-6B fine-tuned on the SocraTeach dataset (Liu et al., 2024a).
- TutorRL-7B: A fine-tuned variant of Qwen/Qwen2.5-7B-Instruct, aligned with pedagogical principles using reinforcement learning (GRPO) in a synthetic multi-turn classroom setting (Dinucu-Jianu et al., 2025).

Note that SocraticLM and TutorRL-7B use the V1 configuration, as we adopt their original training prompts.

¹<https://developers.openai.com/api/docs/models/gpt-4.1-nano>

3.5 Evaluation

We randomly sample 1,000 responses from the test set, equally distributed across datasets, and evaluate them using models from AITutor-EvalKit (Naeem et al., 2025) – namely, GPT-5 and LoMTL. We assess four pedagogical dimensions: *mistake identification*, *mistake location*, *providing guidance*, and *actionability*. Before conducting this evaluation, we perform a factuality check using GPT-5 with the prompt provided in Appendix D. This ensures that we assess pedagogical qualities only for factually correct responses, as responses containing factual errors are considered poor by default.

In addition, we conduct a small-scale human evaluation on an additional randomly sampled subset of the test set.

4 Results & Analysis

4.1 Automatic Evaluation

We first evaluate 1,000 randomly sampled responses from the test set for factuality using GPT-5. This evaluation is intended to identify cases in which a tutor response contains factual errors, such as introducing incorrect numbers from the task definition or hallucinating details about the student’s solution. Table 2 reports the factuality results, while Appendix E provides a breakdown of the error categories assigned by GPT-5 to responses judged as non-factual. As shown in Table 2, GPT-5 achieves the highest factuality scores, whereas GPT-4.1-nano performs substantially worse. While GPT-5 may exhibit some degree of bias when evaluating its own outputs, the high scores may also reflect its strong factual accuracy. The open-source baselines are also considerably less factual overall. Although SocraticLM is factually correct in roughly 50% of cases, our manual analysis suggests that this is partly because it often produces generic follow-up questions about the task or the student’s solution, rather than directly engaging with the specific misconception, thereby avoiding factual errors. For our models, both SFT and DPO generally improve factuality compared to their respective base models. For the Qwen models, DPO often provides additional gains beyond SFT, whereas for GPT-4.1-nano, DPO degrades factuality relative to SFT. For Qwen3-8B, the configuration that conditions on both the correctness flag and the gold solution (V4) yields the strongest factuality results, indicating that access to this additional in-

Model	Overall	Stu. Inc.	Stu. Cor.
<i>Proprietary Models</i>			
GPT-5	96.20	97.49	95.01
GPT-4.1-nano	38.10	15.87	58.54
<i>Existing Tutoring Models</i>			
SocraticLM	50.20	51.17	49.34
TutorRL-7B	39.80	33.40	45.68
<i>Open-Source Base Models</i>			
Qwen3-4B	11.70	3.13	19.58
Qwen3-8B	13.70	2.09	24.38
<i>Qwen3-4B Fine-tuned</i>			
+ SFT V1	43.10	24.01	60.65
+ DPO V1	32.90	24.63	40.50
+ SFT V2	56.10	35.85	74.57
+ DPO V2	58.30	39.04	76.01
+ SFT V3	48.60	27.97	67.56
+ DPO V3	53.20	40.08	65.26
+ SFT V4	55.70	33.82	75.82
+ DPO V4	56.00	35.91	74.47
<i>Qwen3-8B Fine-tuned</i>			
+ SFT V1	46.30	21.50	69.10
+ DPO V1	48.40	42.17	54.13
+ SFT V2	55.60	35.91	73.70
+ DPO V2	39.20	28.39	49.14
+ SFT V3	49.70	26.51	71.02
+ DPO V3	51.50	38.83	63.15
+ SFT V4	59.70	39.67	78.12
+ DPO V4	60.10	42.59	76.20
+ DPO V4*	70.10	56.99	82.15
<i>GPT-4.1-nano Fine-tuned</i>			
+ SFT V4	84.40	75.57	92.51
+ DPO V4	66.70	70.56	63.15

Table 2: Factuality metrics (%) across models. Columns report overall factuality, factuality when the student is incorrect (Stu. Inc.), and when they are correct (Stu. Cor.). Best results per group are in **bold**. Qwen3-8B + DPO V4* denotes the rerun of the DPO stage using the extended dataset.

formation is beneficial. Our manual analysis further shows that the best-performing configuration, Qwen3-8B + DPO V4, still frequently fails by copying an incorrect number from either the task definition or the student’s solution. To address this issue, we automatically created 20,000 additional preference pairs by perturbing the numerical values in the reference responses used during training. We then reran the DPO stage with this extended dataset. This led to an absolute factuality improvement of approximately 10 percentage points, resulting in the final Qwen3-8B + DPO V4* model.

Table 3 presents the results of the automatic evaluation for the pedagogical dimensions of *Mistake Identification (MI)*, *Mistake Location (ML)*, *Providing Guidance (PG)*, and *Actionability (AC)* as assessed by GPT-5 and LoMTL. These metrics are

computed only for responses classified as factually correct in the previous stage. A notable observation is the substantial disagreement between the two evaluators. In particular, GPT-5 assigns significantly higher scores to its own responses than LoMTL does, especially for the Actionability dimension (76.23 vs. 20.49). Manual analysis suggests that GPT-5 responses often reveal the full solution and final answer to the student. While this may appear helpful, it leaves little room for further student engagement. It therefore reduces the student’s opportunity to solve the problem actively, making such responses less actionable from a pedagogical perspective. This behavior may indicate a potential bias of GPT-5 when evaluating its own outputs.

SocraticLM is rated poorly by both evaluators on the MI, ML, and PG dimensions, but receives relatively high Actionability scores from LoMTL (92.08). Our manual analysis shows that SocraticLM frequently produces general follow-up questions about the task, which are actionable because they prompt the student to respond. However, these questions are often not directly related to the student’s specific mistake. Because LoMTL is based on a smaller 2B model, it may fail to detect this lack of targeted feedback, whereas GPT-5 appears better able to capture this issue, resulting in lower Actionability scores (64.58).

TutorRL-7B is consistently rated as highly actionable by both evaluators, but performs poorly on MI and ML according to GPT-5 evaluation.

For GPT-4.1-nano fine-tuning, SFT V4 outperforms DPO V4 across all dimensions under GPT-5. However, under LoMTL, DPO V4 achieves uniformly strong results, with zero “No” cases across all dimensions. Notably, GPT-4.1-nano + DPO V4 is the top-performing model according to LoMTL.

Finally, Qwen3-8B + DPO V4* outperforms existing tutoring models (SocraticLM, TutorRL-7B) and GPT-4.1-nano fine-tuned models across most dimensions under GPT-5, with the exception of Actionability. Under LoMTL, it slightly underperforms GPT-4.1-nano fine-tuned models but still consistently surpasses SocraticLM and TutorRL-7B.

Examples illustrating models outputs are shown in Appendix I.

4.2 Human Evaluation

In addition to automated metrics, we conducted a human evaluation with 10 annotators, including Master’s and PhD students in NLP and professors. The study was designed to test two hy-

potheses: **(H1)** within a fixed backbone, our post-training pipeline improves pedagogical tutoring quality (DPO > SFT > Base); and **(H2)** our best post-trained model is competitive with a strong proprietary baseline.

We randomly selected 35 dialogs from the test set, balanced across dataset sources, and conducted pairwise comparisons. Each annotator evaluated a subset of dialogs (5 shared and 3 unique per annotator), resulting in 105 total pairwise comparisons (15 shared and 9 unique per annotator). For each dialog, we included one response from each of the following models: Qwen3-8B Base, Qwen3-8B SFT V4, Qwen3-8B DPO V4*, and GPT-5. This setup resulted in the following evaluation pairs: **H1a**: Qwen3-8B Base vs. Qwen3-8B SFT V4; **H1b**: Qwen3-8B SFT V4 vs. Qwen3-8B DPO V4*; and **H2**: Qwen3-8B DPO V4* vs. GPT-5. Annotators were shown the dialog context, the reference (gold) solution, and the flag indicating whether the student’s solution was correct. They were then asked to select which response was better or whether both were equally good or poor (see Figure 2 and Appendix J). Based on majority voting and individual annotator judgments over pairwise comparisons, Qwen3-8B SFT V4 was preferred over Qwen3-8B Base in 67.6% of cases, Qwen3-8B DPO V4* was preferred over Qwen3-8B SFT V4 in 35.3% of cases, and Qwen3-8B DPO V4* was preferred over GPT-5 in 54.3% of cases. Inter-annotator agreement, measured using Fleiss κ , was 0.55, indicating moderate agreement.

Analysis of the reasons chosen by annotators shows consistent patterns across comparisons. Qwen3-8B Base is frequently judged as factually incorrect, less clear, and less coherent, and as providing less helpful guidance than Qwen3-8B SFT V4. Similarly, Qwen3-8B SFT V4 is often rated worse than Qwen3-8B DPO V4* for the same reasons. Compared with Qwen3-8B DPO V4*, GPT-5 is most often not chosen for revealing the final answer and providing less effective scaffolding for student reasoning.

We also evaluated the same responses with GPT-5 and LoMTL, again applying factuality filtering first. We then converted the four pedagogical annotations into scalar scores (1 for *Yes*, 0.5 for *To some extent*, and 0 for *No*; non-factual responses were assigned a score of 0 by default), summed these scores, and compared the resulting totals for the same response pairs used in the human evaluation. Under GPT-5-based evaluation, GPT-5 was ranked above Qwen3-8B DPO V4* in

Model	Mistake Ident.			Mistake Loc.			Providing Guid.			Actionability		
	Yes	Some	No	Yes	Some	No	Yes	Some	No	Yes	Some	No
<i>Proprietary Models</i>												
GPT-5 (GPT-5)	94.00	0.86	<u>5.14</u>	81.37	9.42	<u>9.21</u>	90.15	5.35	4.50	76.23	10.28	13.49
GPT-5 (LoMTL)	85.08	1.56	13.36	66.15	3.56	30.29	63.92	12.69	23.39	20.49	9.35	70.16
GPT-4.1-nano (GPT-5)	81.58	6.58	11.84	42.11	26.32	31.58	93.42	3.95	2.63	48.68	39.47	11.84
GPT-4.1-nano (LoMTL)	53.95	1.32	44.74	44.74	1.32	53.95	42.11	7.89	50.00	9.21	5.26	85.53
<i>Existing Tutoring Models</i>												
SocraticLM (GPT-5)	4.58	8.33	87.08	1.67	8.33	90.00	22.92	28.75	48.33	64.58	10.00	25.42
SocraticLM (LoMTL)	5.42	88.33	6.25	5.42	62.92	31.67	15.42	80.00	4.58	92.08	7.08	0.83
TutorRL-7B (GPT-5)	27.50	35.62	36.88	10.62	30.63	58.75	66.88	29.38	<u>3.75</u>	87.50	10.00	<u>2.50</u>
TutorRL-7B (LoMTL)	81.25	14.37	4.38	56.88	31.87	11.25	63.75	33.12	3.12	91.88	7.50	0.62
<i>Qwen3-8B Fine-tuned</i>												
+ SFT V4 (GPT-5)	42.63	23.16	34.21	28.95	25.79	45.26	61.58	22.11	16.32	57.89	27.37	14.74
+ SFT V4 (LoMTL)	68.95	22.11	8.95	51.05	26.32	22.63	61.05	27.37	11.58	81.58	12.63	5.79
+ DPO V4* (GPT-5)	57.14	24.18	18.68	45.05	26.74	28.21	73.63	17.95	8.42	61.54	28.57	9.89
+ DPO V4* (LoMTL)	82.35	12.16	5.49	68.24	20.39	11.37	74.51	19.61	5.88	92.16	5.49	2.35
<i>GPT-4.1-nano Fine-tuned</i>												
+ SFT V4 (GPT-5)	47.24	26.24	26.52	38.12	28.45	33.43	73.20	19.06	7.73	80.11	17.13	2.76
+ SFT V4 (LoMTL)	83.70	12.98	3.31	73.76	16.85	9.39	80.66	16.30	3.04	97.51	1.93	0.55
+ DPO V4 (GPT-5)	34.62	34.62	30.77	27.51	39.35	33.14	46.45	33.14	20.41	38.17	45.56	16.27
+ DPO V4 (LoMTL)	84.62	15.38	<u>0.00</u>	78.99	21.01	<u>0.00</u>	85.50	14.50	<u>0.00</u>	96.75	3.25	<u>0.00</u>

Table 3: Feedback quality metrics (%) across models, evaluated along four dimensions: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. Each dimension reports the percentage of responses rated *Yes*, *To some extent* (Some), and *No*. The evaluator is indicated in parentheses (GPT-5 or LoMTL). **Bold** and **bold italic** denote the highest *Yes* scores, while underline and underline italic denote the lowest *No* scores for the GPT-5 and LoMTL evaluations, respectively. Results for Qwen3-4B and V1-V3 configurations are provided in Appendix H.

most cases, whereas under LoMTL-based evaluation, our model was more often ranked above GPT-5 (see Appendix K). We further computed the average pairwise Cohen’s κ between model-based rankings and human judgments, obtaining 0.44 for GPT-5 and 0.52 for LoMTL. This indicates that LoMTL-based rankings are more consistent with human preferences than those produced by GPT-5.

5 Conclusions

In this work, we compiled a dataset for math mistake remediation by combining existing tutoring datasets with synthetic data derived from math-task datasets containing LLM-generated solutions. We treat these solutions as student responses and generate the subsequent tutor responses with GPT-5; we also generate degraded variants of the GPT-5 responses along several pedagogical dimensions, resulting in a set of preference pairs used for DPO.

We compare different input variants and show that, as expected, providing the model with additional information, such as the gold solution to the task and a flag indicating whether the student’s solution is correct, substantially improves factual correctness. This suggests that intelligent tutoring systems may benefit from being designed not as a

single model responsible for solving the task and checking the student’s work, but rather as a system of multiple components that solve the task, verify the student’s solution, and generate appropriate feedback. This setting is also natural, since human tutors typically have access to the correct solution.

We further show that applying a two-stage post-training pipeline, SFT followed by DPO, with the compiled dataset improves performance according to human evaluation, leading to a preference for our 8B model over a strong proprietary baseline. We also observe improvements for a proprietary model (GPT-4.1-nano): while factual accuracy decreases relative to its SFT variant according to GPT-5, the DPO version performs best on pedagogical dimensions according to LoMTL evaluation. This highlights the generalizability of our approach for pedagogical alignment across models.

Finally, we highlight the challenges of evaluation in this setting, demonstrating that existing automatic evaluators have significant limitations.

Limitations

Our work has several limitations. First, our evaluation focuses on the quality of tutor responses rather than direct learning outcomes. Although *Factual-*

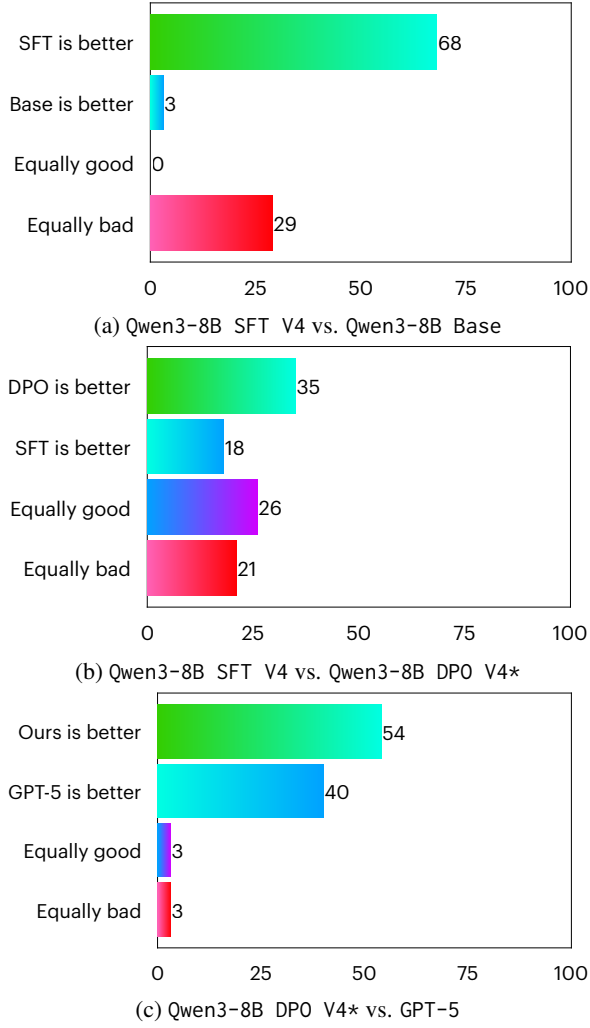


Figure 2: Results of human evaluation (percentages). Four annotators were asked to choose the better response or indicate if both were equally good or bad.

ity, *Mistake Identification*, *Targetedness*, and other aspects are important prerequisites for effective tutoring, we do not evaluate whether students learn better from interacting with such an aligned model. Evaluating learning outcomes would require a controlled user study with real learners, including pre- and post-tests or longitudinal interaction logs. We therefore interpret our results as evidence of improved pedagogical response quality for mistake remediation, rather than as direct evidence of improved educational effectiveness. Notably, most concurrent research on LLM-based tutoring similarly evaluates response quality rather than student learning, highlighting a broader gap in the field.

Second, our evaluation relies in part on LLM-based judges. As demonstrated by our results, different evaluators (GPT-5 and LoMTL) often disagree substantially. This suggests that current automated

evaluators are not yet fully reliable for assessing tutoring quality. While we partially mitigate this through a human evaluation, it remains relatively limited in scale, as the evaluation set is small (105 pairs) and relies primarily on single-annotator judgments, which may introduce noise despite moderate inter-annotator agreement. Similarly, due to resource constraints, the automatic evaluation is conducted on 1,000 test samples only.

Third, the preference data is generated using LLMs. We note that constructing fully human-authored preference data for math mistake remediation is prohibitively costly, as it would require expert tutors to inspect student solutions, identify mistakes, write pedagogically appropriate feedback, and provide controlled lower-quality alternatives across multiple dimensions. Therefore, we view synthetic preference generation as a scalable approximation, but not a substitute for richer human-authored or human-validated data. Moreover, our synthetic preference pairs are generated through minimal revisions produced by two proprietary models from the same family (GPT-5 and GPT-4.1). As a result, the generated data may reflect family-specific stylistic biases and may not fully capture the diversity of natural student-tutor interactions.

Future work should, therefore, focus on several directions. First, it is important to evaluate whether pedagogically aligned models improve learning outcomes for real students. In addition, future work should scale up human and automatic evaluation, incorporate multi-annotator judgments, explore larger backbone models, and design multi-model or human-in-the-loop generation pipelines to improve data diversity and reliability.

Ethical Considerations

The proposed models are trained on synthetic data, which may encode biases in phrasing, tone, or instructional style. As this work is exploratory and the model outputs have not been evaluated with real students, we do not anticipate immediate risks associated with the current study. However, we acknowledge that LLMs may pose risks when deployed in real-world educational settings. In particular, they may generate outputs that appear plausible but are factually incorrect or nonsensical, potentially leading to misguided learning outcomes or reinforcing existing biases. While no direct risks arise from the present work, future applications of these methods in real-world settings should

incorporate appropriate safeguards, such as fact-checking, human oversight, and bias-mitigation strategies.

Acknowledgments

We are grateful to the Google Academic Research Award (GARA) 2024 for supporting this research.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. [A General Theoretical Paradigm to Understand Learning from Human Preferences](#). *Preprint*, arXiv:2310.12036.
- Benjamin S Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16.
- Timothy A Carey and Richard J Mullan. 2004. What is Socratic questioning? *Psychotherapy: theory, research, practice, training*, 41(3):217.
- David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. [From Problem-Solving to Teaching Problem-Solving: Aligning LLMs with Pedagogy using Reinforcement Learning](#). *Preprint*, arXiv:2505.15607.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. [KTO: Model Alignment as Prospect Theoretic Optimization](#). *Preprint*, arXiv:2402.01306.
- 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. *Proceedings of the national academy of sciences*, 111(23):8410–8415.
- Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Jerry Chun-Wei Lin. 2023. Large language models in education: Vision and opportunities. In *2023 IEEE international conference on big data (BigData)*, pages 4776–4785. IEEE.
- Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. CChat-GLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- 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](#). *Preprint*, arXiv:2106.09685.
- Irina Jurenka, Markus Kunesch, Kevin R McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, and 1 others. 2024. Towards responsible development of generative AI for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687*.
- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.
- Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. [Findings of the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors](#). *Preprint*, arXiv:2507.10579.
- 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.
- Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. *arXiv preprint arXiv:2305.18486*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. 2024a. SocraticLM: Exploring socratic personalized teaching with large language models. *Advances in Neural Information Processing Systems*, 37:85693–85721.
- Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J Malan. 2024b. 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*, pages 750–756.

- Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. *arXiv preprint arXiv:2305.14536*.
- Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. [Unifying AI Tutor Evaluation: An Evaluation Taxonomy for Pedagogical Ability Assessment of LLM-Powered AI Tutors](#). *Preprint*, arXiv:2412.09416.
- Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2023. Automated distractor and feedback generation for math multiple-choice questions via in-context learning. *arXiv preprint arXiv:2308.03234*.
- Ethan Mollick and Lilach Mollick. 2024. Instructors as innovators: A future-focused approach to new AI learning opportunities, with prompts. *arXiv preprint arXiv:2407.05181*.
- Numaan Naeem, Kaushal Kumar Maurya, Kseniia Petukhova, and Ekaterina Kochmar. 2025. AITutor-EvalKit: Exploring the Capabilities of AI Tutors. *arXiv preprint arXiv:2512.03688*.
- Benjamin D Nye, Arthur C Graesser, and Xiangen Hu. 2014. AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24:427–469.
- Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. 2024. AutoTutor meets large language models: A language model tutor with rich pedagogy and guardrails. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 5–15.
- 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. *Advances in neural information processing systems*, 36:53728–53741.
- Jiaming Shen, Ran Xu, Yennie Jun, Zhen Qin, Tianqi Liu, Carl Yang, Yi Liang, Simon Baumgartner, and Michael Bendersky. 2024. Boosting reward model with preference-conditional multi-aspect synthetic data generation. *arXiv preprint arXiv:2407.16008*.
- Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard G. Baraniuk. 2024. [Pedagogical Alignment of Large Language Models](#). *Preprint*, arXiv:2402.05000.
- LearnLM Team, Abhinit Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire, Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, and 1 others. 2024. LearnLM: Improving Gemini for Learning. *arXiv preprint arXiv:2412.16429*.
- Rose E Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2023. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. *arXiv preprint arXiv:2310.10648*.
- Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachler. 2021. Are we there yet? – A systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.
- Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. 2023. MR-GSM8K: A Meta-Reasoning Benchmark for Large Language Model Evaluation. *arXiv preprint arXiv:2312.17080*.

A Prompt Templates for Preferred and Non-Preferred Next Tutor Turns

See Figures 3 and 4.

Preferred Tutor Response Generation Prompt (Template)

You are an expert math tutor helping a student understand and fix a mistake in their solution. Your goal is to guide the student toward the correct reasoning without revealing the final answer. Keep your response concise and focused on the core misunderstanding. You will be shown the problem, the student's incorrect solution, and the step where the mistake occurs; produce the next helpful tutor response.

<task_description>

You will be shown:

1. A detailed guideline describing evaluation aspects of a good tutor response.
2. A math problem, the correct (gold) solution, the student's solution, and the specific step where the mistake occurs.
3. A scaffolding level that has been selected for you.

Your task is to produce a high-quality tutor response that helps the student progress meaningfully while adhering to the guideline AND the assigned level.

</task_description>

<guideline>

A good tutor response is evaluated along the following aspects:

1. Factuality + Non-contradiction + No Nonsense: The response should be factually correct, should not contradict what the student has said, and should not contain irrelevant information. Example:

Conversation history: [the task about gift cards]

Bad: If Ms. Jones received 5 gift cards worth \$10 each, then 1/3 of the thank you cards contained a gift card, meaning 3 out of every 5 cards had a gift card, and you can use this information to solve for the total number of students in her class. --> Factually incorrect

Better: That's correct! Now, if 1/3 of the thank you cards contained a gift card, how many thank you cards did she

receive in total?

2. Mistake Identification + Location: The response should identify, either explicitly or implicitly, that there is a mistake in the student's solution. For example, saying "Nice try" would miss this aspect. Example:

Conversation history: Student: yay
Tutor: Next, we have to find the value of 6 times 9 + 2. Tutor: What is the value of 6 times 9 + 2?

Student: 99

Better: That's a great start!

Remember, we need to follow the order of operations (PEMDAS). Let's multiply 6 times 9 first, then add 2.

Bad: Very good try. --> Mistake Identification + Location aspect is missing

3. Targetedness: The response should address the core misconception/misunderstanding of a student. Example:

Conversation history: [the task about eggs]

Worse: Ok, remember there are only 12 eggs to begin with --> doesn't point to any misconception

Better: It looks like you've got a good start, but let's revisit the single yolks calculation - since 5 eggs had double yolks, that means there are actually $12 - 5 = 7$ eggs with single yolks, not 12.

4. Not revealing the final answer: While it is sometimes necessary and acceptable to share the answer to a substep, the tutor should avoid giving away the final answer. Example:

Conversation history: [the task about average speed]

Worse: You're close, but let's re-examine this - if Jason already drove 30 minutes at 60 miles per hour, that means he covered 30 miles, and he still has 90 miles left, so to cover those 90 miles in 1 hour (60 minutes), not 1 hour 30 minutes, he needs to average a speed of $90/1 = 90$ miles per hour. --> Reveals the final answer

Better: You're very close, but remember the question asks for the average speed needed for the *remaining* portion of the drive, not the overall trip.

5. Clarity + Coherence: The tutor's response should be free of awkward, confusing or misleading wording. A good tutor response acknowledges the student's input and connects it to the next step. Example: Conversation history: [the task about gift cards]
Student: She got 5 gift cards since \$50 divided by \$10 is equal to 5.
Worse: That's correct! Now, if 1/3 of the thank you cards contained a gift card, how many thank you cards did she receive in total?
Better: Ok. And if she got 5 thank you cards that had gift cards in them, and these were 1/3 of the total number of thank you card, she got, how many thank you cards did she get total? --> More explicit connection between the student's input and the next step

Scaffolding level:
- Assigned scaffolding level: {level}
- Level notes: {level_notes}
</guideline>

<problem>
{question}
</problem>

<correct_solution>
{correct_solution}
</correct_solution>

<student_solution>
{attempt}
</student_solution>

<erroneous_step>
Error step: {error_step}
Reason: {error_reason}
</erroneous_step>

Read everything carefully.
Use the assigned scaffolding level above when crafting the tutor response-match its tone, pacing, and support budget.

Wrap your output in <response> >...</response> tags.

Figure 3: Prompt template used to generate preferred next tutor turn. Curly-brace fields denote placeholders filled per instance. The scaffolding level and notes used are: *L0 (metacognitive) – tone-first, Socratic nudge, and one check question, with a maximum of two sentences.*

Non-Preferred Tutor Response Generation Prompt (Template)

You are an expert tutor teaching another tutor how not to respond to a student's response .

When revising tutor responses, only focus on the following aspect(s):
{aspects}

Details for the aspect(s):
{aspect_instructions}

Your Task

Below is a task student is solving
:
{question}

The student's incorrect solution is:
{attempt}

The correct (gold) solution to the task is:
{correct_solution}

Error step: {error_step}
Reason: {error_reason}

The ideal tutor's next response is
:
{good_response}

Your job is to minimally revise the tutor's response so that it clearly fails to align with the criteria above.

- * Do not rewrite the response completely.
- * Keep as much of the original wording as possible.
- * Only adjust what's necessary to make the response misaligned.

Put your generated bad response in
<response></response> tags.

Figure 4: Prompt template used to generate non-preferred next tutor turn. Curly-brace fields denote placeholders filled per instance.

B Prompt for Correct Student Solutions

See Figure 5. Example responses include *You got it right. Would you like more practice?* Although we do not specify any particular format in the prompt, we observe that all generated responses consist of two sentences: the first acknowledges the correct-

ness of the solution, and the second asks about next steps. We therefore create combinations by mixing the first and second sentences, yielding 10,000 unique responses.

Prompt for Correct Student Solutions

You are an expert, encouraging math tutor.
Generate exactly 100 distinct one-line tutor responses suitable for cases where a student's solution is correct.
Requirements:

- Each response must be concise, natural-sounding, and encouraging.
- Vary the structure; do NOT start most lines with the same word or pattern.
- Avoid semicolons entirely.
- Avoid overly specific instructions such as asking for diagrams, substitution methods, coding, graphs, etc.
- Follow-up suggestions must be general and applicable across many math topics (e.g., ask if they'd like another problem, another challenge, a summary, questions, etc.).
- Tone should vary: praise, curiosity, invitations to continue, quick check-ins, etc.
- Number each line 1) through 100)
- No bullet points, no quotes, no emojis, no extra commentary.
- Output plain text with exactly ONE response per line and EXACTLY 100 lines.

Figure 5: Prompt template used to generate 100 tutor responses for correct student solutions.

C Prompt Template Used for Training

See Figure 6.

Tutor Response Generation Prompt (Template)

You are a careful, precise, and supportive math tutor.

Your task is to produce the next tutor response in an ongoing dialog.

You will be given:

- 1) The dialog history between the student and the tutor. # V1-V4

- 2) A boolean flag indicating whether the student's solution is mathematically correct. # V2, V4
- 3) A gold solution to the task. # V3, V4

Guidelines for your response:

- If the student's solution is incorrect, guide the student toward the correct reasoning.
- If the student's solution is correct, clearly acknowledge correctness and optionally provide brief reinforcement, intuition, or a natural next step.
- Do NOT invent errors or suggest corrections if the solution is correct.

Your Task

[Dialog History] # V1-V4
{dialog_history}

[Student solution is correct] # V2, V4
{is_correct}

[Gold solution to the task] # V3, V4
{gold_solution}

[Next tutor response]

Figure 6: Prompt template used for training. Curly-brace fields denote placeholders filled for each instance. The # symbol indicates comments included here to show which parts of the prompt are present in different configurations.

D Prompt Template for Factuality Checking

See Figure 7.

Factuality Checking Prompt (Template)

[System Prompt]

You are an expert pedagogical judge.
Your task is to evaluate a math tutor's response to a student based on factuality and categorize any factual issue.

****Factuality**:** The response should not invent numbers, calculations, or details not present in the problem or the student's history.

```

Return fields:
- is_factual: true/false
- student_has_mistake: true/false
- tutor_acknowledged_no_mistake:
  true/false/null
- category: one of
  - student_hallucination: Tutor
    hallucinates something in
    the student's solution
  - task_hallucination: Tutor
    hallucinates something in
    task definition
  - response_incorrect: Tutor's
    feedback/solution is not
    factually correct
  - other: Explain what is wrong
  If is_factual is true, set
  category to null.
- reasoning: brief explanation

Be strict: if any factual issue
exists, set is_factual to false
and choose the best-fitting
category.

Provide your evaluation in the
strictly defined structure.

[User Prompt Template]

[Dialog History]
{conversation}

[Gold Solution to the Task]
{gold_solution}

[Is student's solution correct?]
{is_correct}

[Gold Tutor Response]
{gold_tutor_response}

[Generated Response to Evaluate]
{response}

```

Figure 7: Prompt template used for factuality checking.

E Breakdown of Non-Factual Error Categories

See Table 4.

F Examples of Preference Pairs

Table 5 presents an example of a generated preferred response alongside its degraded variants across different aspects. Compared to the others, the response degraded in *Clarity* is relatively less problematic; it is simply less detailed, making it slightly less clear but still appropriate. Based on this observation, we assign a lower weight to this dimension.

Model	Task H.	Resp. Inc.	Stu. H.	Other
<i>Proprietary Models</i>				
GPT-5	2.63	65.79	23.68	7.89
GPT-4.1	12.66	53.16	31.65	2.53
GPT-4.1-nano	0.81	20.84	77.54	0.81
<i>Existing Tutoring Models</i>				
SocraticLM	60.04	36.35	3.01	0.60
TutorRL-7B	38.04	44.85	15.95	1.16
<i>Open-Source Base Models</i>				
Qwen3-4B	38.39	42.81	18.57	0.23
Qwen3-8B	37.43	50.41	11.59	0.58
<i>Qwen3-4B Fine-tuned</i>				
+ SFT V1	36.73	37.43	22.85	2.99
+ DPO V1	34.87	28.32	32.49	4.32
+ SFT V2	34.85	34.85	23.69	6.61
+ DPO V2	33.81	38.37	24.46	3.36
+ SFT V3	34.82	49.03	14.01	2.14
+ DPO V3	35.26	37.18	22.65	4.91
+ SFT V4	34.09	34.76	27.77	3.39
+ DPO V4	34.09	39.55	21.82	4.55
<i>Qwen3-8B Fine-tuned</i>				
+ SFT V1	40.60	37.80	17.88	3.72
+ DPO V1	30.43	27.33	37.60	4.65
+ SFT V2	32.21	36.71	26.13	4.95
+ DPO V2	14.31	16.12	8.55	61.02
+ SFT V3	40.56	34.99	20.48	3.98
+ DPO V3	40.21	33.81	20.82	5.15
+ SFT V4	31.76	34.49	28.04	5.71
+ DPO V4	31.08	40.60	23.31	5.01
+ DPO V4*	23.08	48.16	21.40	7.36

Table 4: Breakdown of non-factual (NF) error categories (%) across models, identified by GPT-5: task hallucination (Task H.), incorrect response (Resp. Inc.), student hallucination (Stu. H.), and other errors (e.g., inventing a student’s name, typos, or introducing unrelated examples).

G Training Objective

The weighted DPO loss is:

$$\begin{aligned}
\mathcal{L}_{\text{DPO}} = & - \mathbb{E}_{(x_i, y_i^+, y_i^-) \sim \mathcal{D}} \left[w_i \cdot \log \sigma \left(\beta \log \frac{\pi_{\theta}(y_i^+ | x_i)}{\pi_{\text{ref}}(y_i^+ | x_i)} \right. \right. \\
& \left. \left. - \beta \log \frac{\pi_{\theta}(y_i^- | x_i)}{\pi_{\text{ref}}(y_i^- | x_i)} \right) \right] \quad (1)
\end{aligned}$$

where y_i^+ and y_i^- denote the preferred and dis-preferred responses, respectively, and $\beta = 0.3$ (was chosen empirically based on validation performance) controls the strength of the KL penalty. To further stabilize training, we augment the DPO loss with a chosen-response negative log-likelihood (NLL) regularization term:

Dialog history	Tutor: A gasoline tank is $\frac{7}{8}$ full. After 12 gallons have been used, it is half full. How many gallons does this tank hold when it is full? Student: I want to find the capacity of the tank, so I will let x be the number of gallons it can hold when full. Since the tank is $\frac{7}{8}$ full, it has $\frac{7}{8}x$ gallons of gasoline in it. After 12 gallons are used, it has $\frac{7}{8}x - 12$ gallons left, and this is half of the capacity, so $\frac{1}{2}x$. I can set up an equation to relate these expressions: $\frac{7}{8}x - 12 = \frac{1}{2}x$. I want to solve for x, so I will add 12 to both sides and then multiply both sides by 8: $7x - 96 = 4x + 96$, then $3x = 192$. Dividing both sides by 3, I get $x = 64$. This means the tank can hold 64 gallons when full. # Answer 64
Gold solution	Let x represent the number of gallons the tank holds when it is full. We know that the difference between $\frac{7}{8}$ full and $\frac{1}{2}$ full is 12 gallons, so we set up an equation and solve for x: $12 = \frac{7}{8}x - \frac{1}{2}x$, so $12 = \frac{3}{8}x$, and therefore $12 \cdot \frac{8}{3} = x$. Thus, $x = 32$. The tank holds 32 gallons when it is full.
Preferred response	Great setup! After you add 12 to both sides, what exact equation do you have – and when you then multiply by 8, which side gets the +96?
Non-preferred: Factuality	Great setup! After you add 12 to both sides, you should get $7x - 96 = 4x + 96$, so when you then multiply by 8, you get the correct equation. That means your final answer of $x = 64$ is correct.
Non-preferred: Mistake Identification	Nice work! That’s a clever approach.
Non-preferred: Targetedness	Great setup! Remember, you need to carefully combine like terms and isolate x to solve the equation.
Non-preferred: Revealing Answer	Great setup! After you add 12 to both sides, you should get $7x = 4x + 96$ and then $3x = 96$, so $x = 32$. That means the tank holds 32 gallons when full.
Non-preferred: Clarity	Great setup! After you add 12 to both sides, what do you get next? And when you then multiply by 8, what happens?

Table 5: Example of a generated preferred response and corresponding non-preferred responses across different aspects.

K GPT-5 and LoMTL Evaluation on the Human Evaluation Set

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DPO}} + \alpha \sum_i w_i \cdot \mathcal{L}_{\text{NLL}}(y_i^+ | x_i) \quad (2)$$

See Figures 10 and 11.

with $\alpha = 0.005$ (was chosen empirically to stabilize training). Training runs with early stopping based on validation reward margins. After training, the learned adapter is merged into the base model to produce the final inference model.

H Full Automatic Evaluation Results

See Table 6.

I Generation Examples

See Tables 7 to 9.

J Human Evaluation Form

See Figures 8 and 9.

Model	Mistake Ident.			Mistake Loc.			Providing Guid.			Actionability		
	Yes	Some	No	Yes	Some	No	Yes	Some	No	Yes	Some	No
<i>Proprietary Models</i>												
GPT-5 (GPT-5)	94.00	0.86	<u>5.14</u>	81.37	9.42	<u>9.21</u>	90.15	5.35	4.50	76.23	10.28	13.49
GPT-5 (LoMTL)	85.08	1.56	13.36	66.15	3.56	30.29	63.92	12.69	23.39	20.49	9.35	70.16
GPT-4.1-nano (GPT-5)	81.58	6.58	11.84	42.11	26.32	31.58	93.42	3.95	2.63	48.68	39.47	11.84
GPT-4.1-nano (LoMTL)	53.95	1.32	44.74	44.74	1.32	53.95	42.11	7.89	50.00	9.21	5.26	85.53
<i>Existing Tutoring Models</i>												
SocraticLM (GPT-5)	4.58	8.33	87.08	1.67	8.33	90.00	22.92	28.75	48.33	64.58	10.00	25.42
SocraticLM (LoMTL)	5.42	88.33	6.25	5.42	62.92	31.67	15.42	80.00	4.58	92.08	7.08	0.83
TutorRL-7B (GPT-5)	27.50	35.62	36.88	10.62	30.63	58.75	66.88	29.38	<u>3.75</u>	87.50	10.00	<u>2.50</u>
TutorRL-7B (LoMTL)	81.25	14.37	4.38	56.88	31.87	11.25	63.75	33.12	3.12	91.88	7.50	0.62
<i>Qwen3-4B Fine-tuned</i>												
+ SFT V1 (GPT-5)	49.57	15.65	34.78	39.13	19.13	41.74	66.96	13.91	19.13	74.78	11.30	13.91
+ SFT V1 (LoMTL)	66.96	24.35	8.70	56.52	24.35	19.13	57.39	27.83	14.78	79.13	12.17	8.70
+ DPO V1 (GPT-5)	13.56	16.10	70.34	6.78	16.95	76.27	27.97	33.05	38.98	38.98	28.81	32.20
+ DPO V1 (LoMTL)	43.22	44.07	12.71	35.59	34.75	29.66	45.76	39.83	14.41	66.95	22.88	10.17
+ SFT V2 (GPT-5)	39.77	25.73	34.50	27.49	28.07	44.44	63.74	14.62	21.64	58.48	27.49	14.04
+ SFT V2 (LoMTL)	63.89	27.78	8.33	41.67	30.56	27.78	52.78	38.89	8.33	77.78	16.67	5.56
+ DPO V2 (GPT-5)	47.06	17.11	35.83	31.55	22.46	45.99	67.91	14.97	17.11	59.36	26.20	14.44
+ DPO V2 (LoMTL)	68.98	22.46	8.56	57.22	23.53	19.25	63.10	24.06	12.83	81.82	11.76	6.42
+ SFT V3 (GPT-5)	32.84	20.15	47.01	14.93	26.12	58.96	50.75	19.40	29.85	58.96	14.93	26.12
+ SFT V3 (LoMTL)	45.52	32.09	22.39	31.34	29.10	39.55	45.52	30.60	23.88	64.18	15.67	20.15
+ DPO V3 (GPT-5)	57.29	18.75	23.96	42.19	28.12	29.69	72.40	16.15	11.46	64.58	26.56	8.85
+ DPO V3 (LoMTL)	76.56	13.54	9.90	69.27	13.54	17.19	76.04	14.06	9.90	88.54	4.69	6.77
+ SFT V4 (GPT-5)	47.53	20.99	31.48	35.19	23.46	41.36	62.96	23.46	13.58	59.88	20.99	19.14
+ SFT V4 (LoMTL)	71.60	20.99	7.41	58.64	24.69	16.67	65.43	23.46	11.11	80.86	13.58	5.56
+ DPO V4 (GPT-5)	51.16	23.26	25.58	30.23	30.81	38.95	70.93	20.93	8.14	59.88	20.99	19.14
+ DPO V4 (LoMTL)	74.42	17.44	8.14	61.05	23.84	15.12	70.93	16.28	12.79	84.88	8.72	6.40
<i>Qwen3-8B Fine-tuned</i>												
+ SFT V1 (GPT-5)	26.21	26.21	47.57	15.53	27.18	57.28	61.17	19.42	19.42	61.17	21.36	17.48
+ SFT V1 (LoMTL)	58.25	25.24	16.50	53.4	23.3	23.3	68.93	15.53	15.53	79.61	9.71	10.68
+ DPO V1 (GPT-5)	53.96	18.81	27.23	43.56	24.26	32.18	74.75	15.84	9.41	72.28	21.29	6.44
+ DPO V1 (LoMTL)	74.75	19.80	5.45	68.32	16.34	15.35	75.25	16.34	8.42	90.59	6.44	2.97
+ SFT V2 (GPT-5)	42.44	23.84	33.72	26.16	27.33	46.51	63.95	19.77	16.28	52.91	31.40	15.70
+ SFT V2 (LoMTL)	65.12	27.33	7.56	52.33	26.16	21.51	59.88	26.16	13.95	79.07	12.21	8.72
+ DPO V2 (GPT-5)	42.65	22.79	34.56	35.29	19.85	44.76	65.44	15.44	19.12	63.97	18.38	17.65
+ DPO V2 (LoMTL)	73.53	16.91	9.56	50.74	32.35	16.91	63.24	26.47	10.29	80.88	10.29	8.82
+ SFT V3 (GPT-5)	58.27	15.75	25.98	38.58	29.92	31.50	77.17	14.17	8.66	81.10	8.66	10.24
+ SFT V3 (LoMTL)	69.29	22.05	8.66	64.57	21.26	14.17	72.44	18.11	9.45	85.04	7.87	7.09
+ DPO V3 (GPT-5)	54.30	27.42	18.28	43.55	30.65	25.81	75.81	17.20	6.99	74.73	18.28	6.99
+ DPO V3 (LoMTL)	77.42	18.28	4.30	70.97	19.89	9.14	75.81	19.89	4.30	93.01	4.84	2.15
+ SFT V4 (GPT-5)	42.63	23.16	34.21	28.95	25.79	45.26	61.58	22.11	16.32	57.89	27.37	14.74
+ SFT V4 (LoMTL)	68.95	22.11	8.95	51.05	26.32	22.63	61.05	27.37	11.58	81.58	12.63	5.79
+ DPO V4 (GPT-5)	50.00	23.53	26.47	33.82	33.33	32.84	67.16	19.12	13.73	59.80	29.41	10.78
+ DPO V4 (LoMTL)	72.55	18.14	9.31	59.80	24.02	16.18	66.67	25.98	7.35	86.27	8.33	5.39
+ DPO V4* (GPT-5)	57.14	24.18	18.68	45.05	26.74	28.21	73.63	17.95	8.42	61.54	28.57	9.89
+ DPO V4* (LoMTL)	82.35	12.16	5.49	68.24	20.39	11.37	74.51	19.61	5.88	92.16	5.49	2.35
<i>GPT-4.1-nano Fine-tuned</i>												
+ SFT V4 (GPT-5)	47.24	26.24	26.52	38.12	28.45	33.43	73.20	19.06	7.73	80.11	17.13	2.76
+ SFT V4 (LoMTL)	83.70	12.98	3.31	73.76	16.85	9.39	80.66	16.30	3.04	97.51	1.93	0.55
+ DPO V4 (GPT-5)	34.62	34.62	30.77	27.51	39.35	33.14	46.45	33.14	20.41	38.17	45.56	16.27
+ DPO V4 (LoMTL)	84.62	15.38	<u>0.00</u>	78.99	21.01	<u>0.00</u>	85.50	14.50	<u>0.00</u>	96.75	3.25	<u>0.00</u>

Table 6: Feedback quality metrics (%) across models, evaluated along four dimensions: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. Each dimension reports the percentage of responses rated *Yes*, *To some extent* (Some), and *No*. The evaluator is indicated in parentheses (GPT-5 or LoMTL). **Bold** and *bold italic* denote the highest *Yes* scores, while underline and *underline italic* denote the lowest *No* scores for the GPT-5 and LoMTL evaluations, respectively.

Dialog Context	
Tutor: Matt wants cookies for a snack, but his dad wants him to eat healthily. He tells Matt he can have half as many cookies as the number of carrot sticks he eats, plus two extra for cleaning his room. If Matt wants to eat five cookies in total, how many carrot sticks does he have to eat? Student: Step 1: Matt wants to eat five cookies in total. Step 2: His dad tells him he can have half as many cookies as the number of carrot sticks he eats, plus two extra for cleaning his room. Step 3: So, if Matt wants to eat five cookies, he can have $5/2 = 2.5$ carrot sticks. Step 4: But since you can't have half a carrot stick, Matt needs to eat at least three carrot sticks. Step 5: Therefore, Matt needs to eat three carrot sticks in order to eat five cookies. Step 6: ##### 3 Step 7: The answer is: 3	
Correct Solution	
First subtract the two cookies Matt gets for cleaning his room from the total number he wants to eat: $5 - 2 = 3$ Then double the number of cookies to find how many carrot sticks he has to eat: $3 * 2 = 6$. 6	
Model	Response
GPT-5	You've mixed up the relationship in Step 3. The rule is: cookies = $(1/2)(\text{carrot sticks}) + 2$. Set this equal to 5 and solve: $(1/2)c + 2 = 5$ $(1/2)c = 3$ $c = 6$. So Matt needs to eat 6 carrot sticks to have 5 cookies.
SocraticLM	How many carrot strips can Matt have?
TutorRL-7B	It looks like you've made some progress, but there's a small mistake in your reasoning. Let's break it down step-by-step: 1. Matt wants 5 cookies in [garbled text: 89] [garbled text: teacher] Let's start over with the problem. We know Matt can have cookies based on the number [garbled text] (carrot sticks) he eats. Can you write an equation that represents this relationship? The rule is: the number [garbled text] (cookies) Matt gets is half the number [garbled text] (carrots) he [garbled text] + 2 extra for doing chores. If we let c be the number [garbled text] (carrots), can you write what Matt gets in terms of c?
Qwen3-8B SFT V4	Quick check on Step 3—what part of “half as many [carrot sticks] + 2” gives you that equation? Try defining x as the carrot sticks first; then write an expression equaling 5 based only on that definition (and remember those two bonus cookies).
Qwen3-8B DPO V4*	Nice start! Pause on “half as many”: if x carrot sticks means $(x/2) + 2$ cookies equals 5, what equation should you set up before solving it?
GPT-4.1-nano SFT V4	Nice start. Can you write an equation for the number of carrot sticks (call it c) in terms of the cookies Matt eats (call it b)?
GPT-4.1-nano DPO V4	“Plus two for cleaning”: is “half as many ... plus 2” a “Carrots = $(\text{Cookies}/2) + 2$ ” or “Carrots = $(\text{Cookies} - 2) \times 2$ ”?

Table 7: Example of model responses. The [garbled text] markers highlight instances of corrupted or nonsensical output (e.g., mixed-language tokens) produced by the model.

Dialog Context	
Tutor: Max has 8 children and each of his children has the same number of children as he does except for 2 who have 5 children each. How many grandchildren does he have? Student: Max has $8-2 = 6$ children who have $8-1 = 7$ children each. So, the total number of grandchildren from these 6 children is $6 \times 7 = 42$. The 2 children who have 5 children each will contribute an additional $2 \times 5 = 10$ grandchildren. Therefore, Max has a total of $42+10 = 52$ grandchildren. 52 Teacher: Hi Michael, can you guide me through your solution? Student: Sure. I started by subtracting 2 from 8 to get the number of children Max has that have the same number of children as he does. Then I subtracted 1 from that number to get the number of grandchildren each of those children have. I multiplied that number by 6 to get the total number of grandchildren from those 6 children. Then I added the two children who had 5 children each, which gave me an additional 10 grandchildren. So, the total number of grandchildren is $42 + 10 = 52$.	
Correct Solution	
Out of his 8 children, 2 have only 5 children so the rest who have 8 (the same number as him) are $8-2 = 6$ 6 of Max's children have 8 children each for a total of $6 \times 8 = 48$ grandchildren 2 of Max's children have 5 children each for a total of $5 \times 2 = 10$ grandchildren In total he has $48+10 = 58$ grandchildren 58	
Model	Response
GPT-5	You've got the right structure, but there's one key misread: "the same number of children as he does" means those 6 children each have 8 children, not 7. There's no reason to subtract 1. - 6 children have 8 grandchildren each: $6 \times 8 = 48$ - 2 children have 5 grandchildren each: $2 \times 5 = 10$ Total grandchildren: $48 + 10 = 58$.
SocraticLM	How many children does Max have?
TutorRL-7B	That's a great start, Michael! Let's break it down step-by-step to ensure we're on the right track: 1. Max has 6 children with 8 children each. 2. You correctly calculated that each of these 8-child families contributes 6×7 grandchildren. 3. The other 2 children each have 3 children (since they don't have 7 like the others). Now, let's see if we can simplify or double-check any part of your calculation. Can you calculate the number [garbled text] grandchildren from the 6 children first without adding the 2 children yet?
Qwen3-8B SFT V4	What do we know about how many children max has?
Qwen3-8B DPO V4*	Let's start with this part "I subtracted one from that [number]". Why did you do that?
GPT-4.1-nano SFT V4	That's a good start. Can you tell me how many children Max has who have the same number of children as he does?
GPT-4.1-nano DPO V4	"Same number as he"—whom does "he" refer to—Max or one child—and why "-1"?

Table 8: Example of model responses. The [garbled text] markers highlight instances of corrupted or nonsensical output (e.g., mixed-language tokens) produced by the model.

Dialog Context	
Tutor: Tommy is fundraising for his charity by selling brownies for \$3 a slice and cheesecakes for \$4 a slice. If Tommy sells 43 brownies and 23 slices of cheesecake, how much money does Tommy raise? Student: Step 1: Tommy sells 43 brownies for \$3 each, so he raises $43 \times \\$3 = 129$ from the brownies. Step 2: Tommy sells 23 slices of cheesecake for \$4 each, so he raises $23 \times \\$4 = 92$ from the cheesecake. Step 3: In total, Tommy raises $\\$129 + \\$92 = \\$221$. Step 4: ##### 221 Step 5: The answer is: 221	
Correct Solution	
Student's solution is correct	
Model	Response
GPT-5	Correct! Your calculations are spot on: - Brownies: $43 \times 3 = 129$ - Cheesecake: $23 \times 4 = 92$ - Total: $129 + 92 = 221$ dollars Nice work including each step and the units.
SocraticLM	How much money did Tommy raise from the Brownies?
TutorRL-7B	Great job working through the steps, Tommy! You've calculated the money raised from brownies[garbled text] separately. Let's take a small step back to make sure we're on the right track. Can you tell me how much Tommy makes from selling one brownie and one slice of cheesecake?
Qwen3-8B SFT V4	That's right on. Want another try?
Qwen3-8B DPO V4*	Nicely completed. Would you like more practice?
GPT-4.1-nano SFT V4	Well done. Want a quick summary?
GPT-4.1-nano DPO V4	"Slices"— did "23 slices of cheesecake" mean " $23 \times \$4$ " or " 4×23 "?

Table 9: Example of model responses. The [garbled text] markers highlight instances of corrupted or nonsensical output (e.g., mixed-language tokens) produced by the model.

Tutor Response Evaluation

 Welcome to the Tutor Response Evaluation

Please read the following tutor–student conversations. For each conversation, you will see the correct solution to the task, as well as a flag indicating whether the student’s solution is correct. You will also be shown two possible next tutor responses.

Your task is to choose the response that seems best. When deciding, put yourself in the student’s shoes — which response would you find more helpful and easier to understand? You’ll also be asked to choose a reason why one response is better than the other (you can choose multiple options).

It is possible that both responses are generally good, but we still ask you to indicate which one you find better overall, if possible.

To allow you to revisit and, if needed, edit your annotations later, this form will save your email address. This information will only be used for this purpose and will not be used for anything else.

Each conversation appears three times, with different pairs of tutor responses.

You will evaluate eight conversations. The estimated time to complete the form is approximately one hour and twenty minutes.

★ A strong tutor response should:

- Be factually correct. It should not contradict the student’s answer(s) or include irrelevant information. If both responses lack these qualities, they should be considered equally bad. Minor inaccuracies that do not significantly affect the correctness of the response may still be acceptable.
- Explicitly or implicitly identify a mistake in the student’s solution. For example, simply saying “Nice try” does not meet this requirement.
- Focus on the student’s core misconception or misunderstanding.
- Avoid revealing the final answer (although sharing the result of a substep may sometimes be necessary and acceptable).
- Address the misunderstanding step by step by providing both guidance and scaffolding: guidance means helping the student understand what to think about next (e.g., pointing to a relevant concept or strategy), and scaffolding means breaking the problem into manageable steps using structured hints or targeted questions — rather than directly giving away the full solution.
- Be free of awkward, confusing, or misleading wording. A strong tutor response acknowledges the student’s input and clearly connects it to the next step.
- Not invent errors or suggest corrections if the student’s solution is already correct.

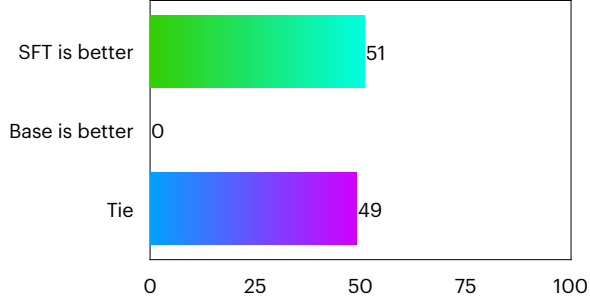
Figure 8: Introduction page of the human evaluation form.

Why did you choose this answer? *

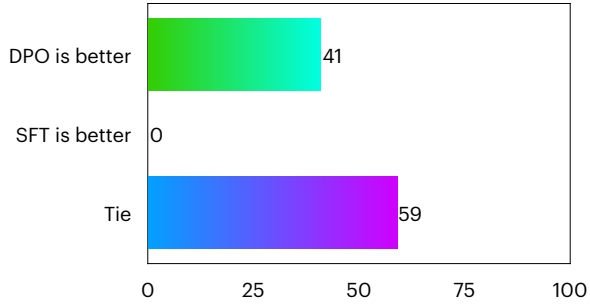
Note: The reasons are ordered from most severe (top) to least severe (bottom), so you do not need to treat all issues as equally serious. Please select only the options that directly influenced your decision when choosing one response over the other. If both responses are equally bad, there is no need to select options referring to a single response.

- ☐ Both responses are poor because they are factually incorrect.
- ☐ Response A is factually incorrect.
- ☐ Response A fails to correctly identify the mistake in the student's solution.
- ☐ Response A provides less effective scaffolding.
- ☐ Response A does not directly address the core misunderstanding.
- ☐ Response A reveals the final answer.
- ☐ Response A is less clear or less coherent.
- ☐ Response A offers less helpful guidance.
- ☐ Response B is factually incorrect.
- ☐ Response B fails to correctly identify the mistake in the student's solution.
- ☐ Response B provides less effective scaffolding.
- ☐ Response B does not directly address the core misunderstanding.
- ☐ Response B reveals the final answer.
- ☐ Response B is less clear or less coherent.
- ☐ Response B offers less helpful guidance.
- ☐ Both responses are equally good because they address the student's misunderstanding equally effectively.
- ☐ Both responses are equally good because they correctly acknowledge that the solution is correct.
- ☐ I skipped this dialog because I believe the provided correct solution is incorrect or misleading.
- ☐ Other (please explain in the comments)

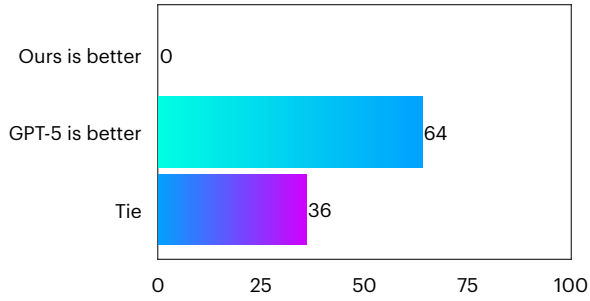
Figure 9: List of reasons provided to human evaluators for selecting a preferred response or indicating that both responses are equally good or bad.



(a) Qwen3-8B SFT V4 vs. Qwen3-8B Base

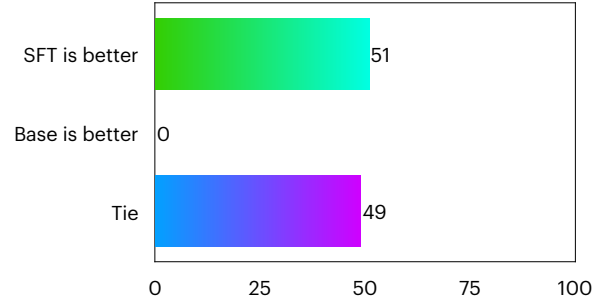


(b) Qwen3-8B SFT V4 vs. Qwen3-8B DPO V4*

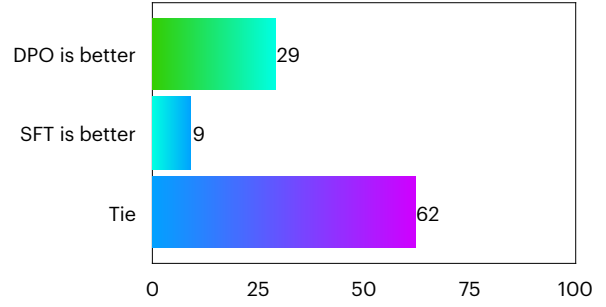


(c) Qwen3-8B DPO V4* vs. GPT-5

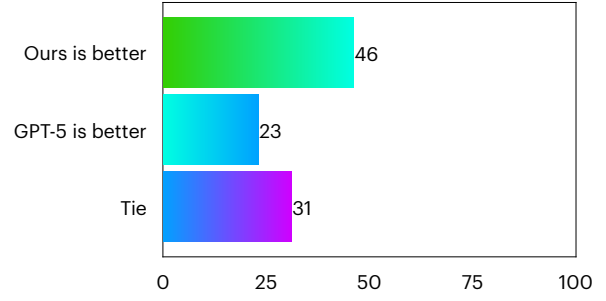
Figure 10: Results of **GPT-5** evaluation (percentages) on the human evaluation set. We convert the four pedagogical annotations into scalar scores (1 for *Yes*, 0.5 for *To some extent*, and 0 for *No*; non-factual responses are assigned a score of 0), sum these scores, and compare the resulting totals across response pairs.



(a) Qwen3-8B SFT V4 vs. Qwen3-8B Base



(b) Qwen3-8B SFT V4 vs. Qwen3-8B DPO V4*



(c) Qwen3-8B DPO V4* vs. GPT-5

Figure 11: Results of **LoMTL** evaluation (percentages) on the human evaluation set. We convert the four pedagogical annotations into scalar scores (1 for *Yes*, 0.5 for *To some extent*, and 0 for *No*; non-factual responses are assigned a score of 0), sum these scores, and compare the resulting totals across response pairs.

Challenges in Machine Translation of Interactive Multimodal Exercises

Lucie Poláková, Miroslav Hrabal, Věra Kloudová,
Michal Novák, Mariia Anisimova and Martin Popel

Charles University
Faculty of Mathematics and Physics
Prague, Czech Republic
{surname, mnovak}@ufal.mff.cuni.cz

Abstract

This paper describes linguistic and technological challenges encountered within an applied project aimed at expanding a large e-learning portal from its original Czech to three other languages: Ukrainian, English and German. Although there seems to be a general belief that machine translation is a solved task in 2026, we show that translating educational content, which in our case is highly terminological, multimodal, interactive and encoded in XML, brings along many challenges of different types, some easily solvable and some not. We also compare our results from the early phase of the project (Transformer-based machine translation) with those after the switch to the LLM-based translation methods. We show that both MT methods are prone to different types of errors, some of which are quite new (such as the undesired correction of counterfactual statements) and require new ways of handling them. The resulting four-language edition of the educational web portal will be freely available to educators, students and researchers by the end of 2026.

1 Introduction

Online learning and hybrid learning, which integrate traditional textbooks with online practice, are becoming increasingly widespread. In addition to many other products and services, major textbook publishers nowadays routinely link their new print textbooks to supplementary material and exercises available online, often via QR codes (e.g. Nadziro et al., 2023, Kempe and Grönlund, 2019). In the Czech educational ecosystem, this gradual shift is occurring alongside a notable increase in non-native speakers within the school system, particularly following the recent arrival of Ukrainian pupils and students (Popel et al., 2024). Ensuring that every learner can access education in a language they understand, at least before they are capable of education in the language of the host

country, is a pressing concern and a key technological challenge. Machine translation (MT) offers a promising way to extend the reach of educational content, though it still faces practical limitations.

In this paper, we explore a case study based on an product development project where multiple translation methods were applied to interactive web exercises, examining the advantages and constraints of these approaches in a real-world educational setting. In addition to Ukrainian as a target language, we focus on English and German as the two most frequent second languages in Czechia.

The paper has two goals: (i) to report and analyze the results of the project that have not yet been published, which in particular include the **switch** from traditional MT methods **to methods based on large language models (LLMs)**; to compare these results on Ukrainian; to report and analyze the results on English and German; and (ii) to describe and analyze the challenges of the chosen task, the key feature of which is **the translation of XML that encodes interactive web exercises with multimodal content**. We identify and classify the linguistic/translatological challenges as well as the technical ones, determine their origins and suggest possible solutions. The description of our experiments could provide valuable insights to the EdTech community about the recent pros and cons of using open source LLMs in domain-specific translation tasks.

The paper is structured as follows: §2 reviews recent developments in the field of MT. §3 describes the data and §4 presents the methodology we employ. In §5, we assess translation quality using automatic metrics, followed by human evaluation. §6 forms the core of the paper, providing a classification and a discussion of the challenges related to our task, along with examples. §7 concludes the paper. Appendix A presents additional examples of mistranslations organized by type; Appendix B offers details for the in-image translation task.

2 Machine Translation

The MT field has undergone a fundamental architectural shift, transitioning from the long-standing dominance of encoder-decoder Transformers (Vaswani et al., 2017) to the adoption of decoder-only Large Language Models (LLMs). Historically, specialized encoder-decoder systems, such as the No Language Left Behind (NLLB) model (NLLB Team et al., 2022), were preferred for their ability to be trained on specific, non-English parallel corpora. For language pairs like Czech-Ukrainian, these dedicated systems achieved superior results by performing direct translation, thereby avoiding the translationese and loss of gender and politeness information associated with traditional pivoting through English (Popel et al., 2024). This architectural specificity allowed for high-fidelity mappings in low-resource or linguistically distant directions and domains that general-purpose models often struggled to capture.

However, recent findings from the WMT25 General Machine Translation Shared Task (Kocmi et al., 2025) indicate that the performance gap between dedicated NMT systems and LLMs has largely closed. Modern LLMs now consistently place in the winning clusters across a vast majority of language directions.¹ The historical advantage of encoder-decoder systems is rapidly vanishing as modern LLMs become natively multilingual, encoding diverse linguistic structures within a unified semantic space during pre-training.

Current research now focuses on refining these LLM-based systems through advanced alignment techniques that go beyond standard Supervised Fine-Tuning (SFT). A notable development is Contrastive Pairwise Optimization (CPO), which trains models to distinguish between gold-standard references and translations that are adequate but imperfect (Xu et al., 2024). By leveraging preference data rather than simple maximum likelihood estimation, CPO enables moderate-scale LLMs to match or exceed the performance of both proprietary models and traditional competition winners.

Translating structured educational content necessitates the high-fidelity preservation of formatting markup, such as HTML, XML or DOCX. This

task is sometimes called *label projection*, especially when used for the cross-lingual transfer of span-annotated NLP datasets with relatively simple labels (i.e. formatting markup) from a high-resource to a low-resource language. There are two main approaches to the task. The first approach involves the joint translation of text and labels, a method that has gained significant traction with the advent of LLMs. By embedding labels, such as XML tags (Thennal et al., 2026) or square brackets (Chen et al., 2023), directly into the input sequence, models learn to generate formatted output in a single pass. While effective for simple entity markers, these joint systems often struggle with the structural integrity required for complex markup, such as nested online exercises in XML formats. Conversely, the second approach, which we use, utilizes a modular two-step pipeline where the plain text is translated first, followed by a separate projection of the labels using word alignment or forced decoding (Parekh et al., 2024). This decoupled strategy remains robust for contemporary localization workflows for structured documents (often using the XLIFF format and Translation Management Systems) as it allows for arbitrary pre- and post-processing of the text and labels independently, ensuring that the structural validity of the markup is maintained regardless of the translation model’s complexity (Zenkel et al., 2021).

3 Data

The educational web portal we work with, *Škola s nadhledem*,² currently contains more than 9,000 interactive exercises designed for practicing curriculum material from the first years of primary school through to the final high-school year. They cover a wide range of school subjects, including foreign languages and computer science. In terms of interactivity, the exercises include many different types, such as drag-and-drop tasks, yes/no quizzes, multiple-choice questions, gap-filling activities, memory games, crosswords, mind maps, timelines, etc. They also work with various media, so in addition to text, they may contain images, audio and other elements. In the past three years, the portal has attracted over 1.5 million users, with more than 100,000 students practicing on it each month. It is also part of the hybrid approach to educational materials, combining printed textbooks or workbooks with interactive exercises that provide

¹In WMT25, LLMs were the predominant approach, used by all but one external submission (Kocmi et al., 2025). Overall, the best-performing model in the human evaluation is Gemini 2.5 Pro [...]. It places in the top cluster for 14 of the 16 evaluated language pairs and is on par with or surpasses human translation in 10 of those pairs. (Kocmi et al., 2025).

²<https://skolasnadhledem.cz>

immediate feedback. Users, regardless of whether they use the company’s printed textbooks at school, can freely practice topics of their choice.

Some of the exercises are not suitable for translation. For example, reading exercises for the youngest learners, such as building words out of letters or syllables, anagram-based games or cross-words. The selection of content for translation was therefore based on the general suitability of each exercise for translation across different subjects and grade levels. So far, we have worked with portal content covering approximately 1,000 preselected exercises from subjects like history, geography, math, physics, biology, chemistry, civics, etc., and we are gradually expanding this scope by adding other suitable thematic areas.

From the translation perspective, the web portal data is highly specific: exercises are typically very short, rarely provide multi-sentence context, include many domain-specific terms, and are formatted in very complex HTML, with images or other elements. These factors make the task challenging even for state-of-the-art MT systems.

Standard MT evaluation requires reference data, i.e., human translations for comparison. We therefore created an evaluation dataset by manually translating 396 selected exercises from biology, chemistry and geography, split into a development set (190 exercises) and a test set (206 exercises). Both sets were translated into Ukrainian, English, and German by professional translators.

4 Methodology

4.1 Collecting data for domain adaptation

In order to adapt our system for translation, we need to obtain parallel bilingual and monolingual in-domain data in the target language. Parallel data can be used directly during fine-tuning, while monolingual data must be back-translated into the source language first.

For Czech-English, we used two data sources: publicly available parallel corpus CzEng 2.0 (Kocmi et al., 2020) and back-translated excerpts from English and Czech Wikipedia.

For Czech-Ukrainian, we used the dataset collected in our previous work (Poláková et al., 2025).

For Czech-German, we identified several online sources that could contain texts from the educational domain (e.g. www.goethe.de, www.oegp.cz, www.rkfpraha.cz). Using the Bitextor tool (Esplà-Gomis, 2009), we downloaded publicly available

pages from these websites, extracted the texts, aligned them based on dictionary and metadata, and then segmented, paired, and filtered them at the sentence level. The result of this process was a dataset containing 57,000 sentence pairs. The monolingual data were extracted from www.oebv.at using custom scripts tailored to this source (targeted textbook collection, filtering out foreign-language textbooks), obtaining 1.2 million German sentences from the educational domain. We further augmented the data with back-translated excerpts from German and Czech Wikipedia.

4.2 Model adaptation

We used a locally installed EuroLLM-9B-Instruct LLM (with 9 billion parameters) for translation from Czech into all three target languages. As full fine-tuning of such a large model would be computationally very demanding, we used the *Low-Rank Adaptation* (LoRA) method for fine-tuning, in which most of the model’s weights remain frozen and smaller adapters, consisting of an order of magnitude fewer parameters, are trained for individual linear layers of the model.

We used three different methods for fine-tuning the models: Supervised Fine-Tuning (SFT), Contrastive Preference Optimization (CPO, Xu et al., 2024) and Simple Preference Optimization (SimPO, Meng et al., 2024). For SFT, we simply used the collected parallel data. For CPO and SimPO, parallel data alone are insufficient; a preference dataset consisting of triplets (source text, preferred translation, non-preferred translation) is required. We therefore also translated the parallel data using the model after the SFT phase. We selected the preferred translation using a combination of publicly available *quality estimation* models (MetricX, Juraska et al., 2024) and publicly available LLMs (Gemma-3-27B-it). Preliminary experiments confirmed that CPO and SimPO bring the best results when applied on top of SFT (we coin these systems SFT+CPO and SFT+SimPO, respectively).

5 Results

The experiments were evaluated intrinsically using automated metrics (§5.1) and human evaluation (§5.2), based on manually translated test sets. The system selected for publication was then evaluated extrinsically within the web portal for all three languages (§5.3). The project is primarily an

applied product-development effort rather than a purely research-oriented study, hence the evaluation aim was to develop a functional educational product and identify exercises suitable for real deployment in the target platform. From this perspective, the evaluations described below thus focused on practical criteria such as pedagogical usefulness and production readiness, serving product-selection and quality-assurance rather than establishing annotator-independent ground truth.

5.1 Automatic Metrics

Table 1 shows the automatic evaluation on the development sets for each language pair using two automatic metrics: chrF2 (Popović, 2015) and MetricX24 (Juraska et al., 2024).³ System B using SFT+CPO was evaluated as the best one in all three language pairs according to both automatic metrics.

5.2 Intrinsic Manual Evaluation

In the intrinsic manual evaluation, the annotators saw side by side the source Czech text, translations from three (or four, in the case of Ukrainian) MT systems (anonymized and presented in random order) and the human reference translation. The evaluation focused primarily on selected, more difficult or longer passages – i.e., those that were linguistically complex. Experts in the target language (native speakers for Ukrainian) evaluated these segments on a 0–10 scale (0 being the worst, 10 being the best). Since the goal of this evaluation phase was to select the best-performing system, the evaluators skipped lines where they would assign the highest scores (10) to all three systems. As a result, evaluation results cannot be used to determine the absolute performance of individual systems (nor to compare across languages); rather, they represent a relative comparison of the systems against one another.

For Ukrainian (Table 2), we compared a standard encoder-decoder NMT system (the pre-LLM Charles Translator, Popel et al., 2024) with the following three systems: SFT (A), SFT+CPO (B), and EuroLLM-9B-Instruct (C) on 1800 segments. The three LLM-based systems are substantially better than the pre-LLM system; SFT+CPO is significantly the best system ($p < 0.05$, paired t-test).

For German and English (Table 3), all translations were of very high quality, and the results of

the individual systems did not differ significantly from one another in either language. Approximately 500 segments were evaluated for each language. For German, we used systems A, B and C (we have not trained a Pre-LLM system). For English, we replaced A (which had the lowest score in German) with a newly trained SFT+SimPO system (D). Systems B and C were the best, with no significant difference. We have selected system B (SFT+CPO) for deployment for both languages.

5.3 Extrinsic Manual Evaluation

The extrinsic human evaluation phase consisted of testing the functionality of 540 translated exercises within the web portal. The procedure was quite straightforward: the evaluator “played the game” and tested several aspects of the exercise at once: translation quality and functionality within the context of the given exercise/task, consistency and technical aspects such as switching among the four languages at any step, correct visualization, correct exercise scoring and display of the correct solution (and also correct solution explanation, where present). There was an easy three-way division: *good* (the exercise has no issues, it works well and can be directly published), *okay* (there are some minor issues that do not affect the functionality of the exercise and mostly can be fixed easily,⁴ see §6), *bad* (the exercise has serious issues and cannot be published in its current form)⁵. Additionally, there was a *not applicable* option for exercises where it was determined that their translation does not make sense, e.g., completing morphological suffixes in Czech, disambiguating homonyms, etc.

In this way, typical patterns of dysfunction were revealed, categorized and they are gradually solved. After the functionality of the exercises is refined, a subset that is already working well is deployed to the production portal accessible to the public. This process is iterative.

The results of the extrinsic human evaluation in Table 4 show that 90% of the translated exercises were evaluated as either *good* or *okay* for Ukrainian, while for English and German, the rates were 75% and 69% respectively. However, the results are not directly comparable across languages, as the translations were evaluated by different evaluators.

⁴such as non-standard use of upper- and lowercase letters, punctuation or special signs, some minor inflection errors

⁵incorrect translations, missing translations or other technical issues (see above) that would prevent the user from successfully completing the exercise

³Specifically, we used the QE version of <https://huggingface.co/google/metricx-24-hybrid-xl-v2p6>, without the “reference” parameter.

System	German		English		Ukrainian	
	chrF2↑	MetricX24↓	chrF2↑	MetricX24↓	chrF2↑	MetricX24↓
Pre-LLM Charles Translator					63.1 ± 0.8	3.73 ± 0.11
A: SFT	65.6 ± 0.7	3.30 ± 0.10	68.5 ± 0.7	5.04 ± 0.10	62.1 ± 0.8	3.59 ± 0.08
B: SFT+CPO	67.1 ± 0.7	2.90 ± 0.09	73.1 ± 1.0	4.68 ± 0.10	64.6 ± 0.7	3.33 ± 0.08
C: EuroLLM-9B-Instruct	65.6 ± 0.8	3.15 ± 0.10	71.6 ± 0.7	4.80 ± 0.10	63.5 ± 1.0	3.59 ± 0.09
D: SFT+SimPO	–	–	72.9 ± 0.7	4.78 ± 0.10	–	–

Table 1: Translation quality measured with automatic MT metrics chrF2 (↑ = higher is better, with 95% CI intervals calculated using 1000 bootstrap samples) and MetricX24 (↓ = lower is better). The best score is in **bold**.

System	Ukrainian↑
Pre-LLM Charles Translator	8.20 ± 2.6
A: SFT	8.75 ± 2.2
B: SFT+CPO	8.81 ± 2.1
C: EuroLLM-9B Instruct	8.68 ± 2.2

Table 2: Human evaluation results for Czech to Ukrainian, average on the scale 0–10 and standard deviation. The significantly ($p < 0.05$) best score is in **bold**.

Quality	DE (%)	EN (%)	UK (%)
good	43	46	54
okay	26	29	36
bad	25	17	4
not applicable	7	7	6

Table 4: Percentage distribution in human evaluation of the demo portal for German (DE), English (EN), and Ukrainian (UK). Evaluated exercises in total: 522 (DE), 523 (EN), 540 (UK).

System	German↑	English↑
A: SFT	7.81 ± 3.2	no eval
B: SFT+CPO	7.96 ± 3.1	8.16 ± 3.1
C: EuroLLM-9B-Instruct	7.95 ± 3.1	8.19 ± 2.9
D: SFT+SimPO	not used	8.00 ± 3.1

Table 3: Human evaluation results for Czech to German and Czech to English, average on the scale 0–10, higher is better ↑. The best score and other scores that are not significantly ($p < 0.05$) worse are in **bold**.

6 Analysis of the Results/Challenges

The results of the manual evaluations (§5.2 and §5.3) show that the translation systems perform well in less contextualized translation scenarios. In the educational domain, these are usually topics related to society, family, culture, and traditions. Another aspect is text complexity: as expected, translation quality was higher for less advanced exercises. In fact, it was very high and these exercises were ready to use (no further editing) even for more specialized topics such as mathematics, history, geography, etc. The relatively low proportion of well functioning exercises in German (see Table 4) is most likely attributed to its typological specificity in connection with a special type of exercise (filling gaps), see §6.3.

6.1 Typical errors in pre-LLM and LLM-based MT systems

Based on our extensive experience in evaluating traditional MT systems, we have a good understanding of the typical errors made by machine translators before the LLM era. Also, many studies have already examined machine translation outputs, focusing on the typology of errors or the typology of changes made by post-editors to the output of machine translation, e.g. Kloudová et al. (2021).

Typical errors in standard MT include failing to distinguish between homonyms in the given context, inconsistency in using gender, spelling of foreign words adopted from the original language, improper collocations, preserving the number of nouns (plural vs singular) and word-to-word translations that do not reflect the meaning of the original, to name just the most common ones.

According to the evaluations performed, new types of errors were identified that differ in certain aspects from those made by standard MT. The following subsections provide a comprehensive, data-driven classification of these errors.

6.2 Lexical challenges

6.2.1 Translation of domain-specific terms

A quite significant type of error identified in translations to German and English, similarly to errors de-

tected earlier in the Ukrainian translation (Poláková et al., 2025), is the translation of domain-specific terms. See Example 1 or Appendix A. This problem occurred primarily in biology. These terms are often unfamiliar not only to the average native Czech speaker but also to professional translators without specific knowledge of the given scientific field.⁶

- (1) **SRC_{cs}**: *ortorula*
MT_{en}: orthorhombic
HT_{en}: orthogneiss
MT_{de}: Orthorula
HT_{de}: der Orthogneis

Currently, our system cannot fully handle these cases. We have been able to replicate these terminology errors in commercial machine translation systems and large language models such as Google Translate, DeepL and ChatGPT up to its free version 5 mini. In order to achieve a more successful translation of domain-specific terminology, we are currently experimenting with different approaches.

We are extending the system with optional terminology constraints via external glossaries. Glossaries allow us to curate in-domain terms independently and quickly introduce missing entries identified during human evaluation, without the need to retrain the underlying model. Building these glossaries at scale remains an open engineering task, and we are exploring sources such as bilingual titles from WikiData and term pairs mined from our parallel training data.

6.2.2 Homonymy

Our predominantly sentence-level (or, due to the nature of our data, sometimes phrase-level) segmentation can reduce the ability to disambiguate homonymous expressions. This is a broad issue with impact on the whole vocabulary we translate, including terminology. Solving lexical homonymy without a sufficient context is known to be an impossible task in theory. At the same time, lexical errors are particularly serious because they substantially change meaning and impede understanding. Very often, because of a single lexical error, the whole exercise is not functional, e.g. in Example 2:

- (2) **SRC_{cs}**: *Má svíčku.*
MT_{en}: She has a candle.
HT_{en}: It (the engine) has a spark plug.

Dealing with homonymy each time anew also sometimes leads to translation inconsistencies within one exercise. For instance, the Czech word *předmět* has multiple meanings (mostly *object*), but there is a high probability it should keep the same meaning within each exercise. Nevertheless, several different translations of this word were documented in the English version of one exercise: *object, subject, item, product*. Similarly, in the German version: *Schulprojekt, Objekt, Gegenstand*.

To mitigate these issues, we are planning to modify the translation process to operate on larger chunks of exercise segments jointly. We expect this to enable the model to select the appropriate semantic variant of a word based on context. According to our preliminary small-scale experiments on several such examples, this indeed does help; however, we need to conduct additional testing to ensure that this does not degrade the quality of the translations in different ways.

6.2.3 Non-existent words

Words that do not exist in the target language (e.g. non-existent compounds, non-standard spellings, see Example 3 or Appendix A) have been encountered in standard MT. Their impact on readers' comprehension of the text was examined (Macken et al., 2019). Based on our analysis of exercises translated by the new LLM-based systems, we identified a relatively large number of newly created words in a comparatively small sample of texts.

- (3) **SRC_{cs}**: *valcha*
MT_{en}: valcharium
HT_{en}: washboard

However, a quantitative study confirming a higher incidence of these inventions, or their different structure (see the Latin suffix in Example 3), in LLM-based translation has yet to be conducted.

6.2.4 Mistranslations based on form similarity

Another recurring phenomenon are mistranslations that apparently arose due to the similarity in form between unrelated words (Examples 4–5).

- | | |
|---|--|
| (4) SRC_{cs} : <i>kopí</i> | (5) SRC_{cs} : <i>žně</i> |
| MT_{en} : kopy | MT_{en} : gently |
| HT_{en} : spear | HT_{en} : harvest |
| MT_{de} : kopieren | MT_{de} : herzlich |
| HT_{de} : Speer | HT_{de} : Ernte |

In these cases, the incorrect forms appear to have been triggered by superficial similarity: Example 4 by confusion with *kopírovat* (*to copy*), Example 5

⁶In the following examples, we use **SRC_{cs}** for the source Czech text, **MT** for machine translation by the published SFT+CPO system and **HT** for the reference human translation.

by interference from the Czech *něžně* (gently). For more such cases, see Appendix A.

6.2.5 Untranslated words

In some cases, the translation failed for very specific groups of words, which remained untranslated. It is words that are in the source language: (i) very short (*var* – *boiling point*, *sup* – *vulture*), (ii) across-language homographs (*city* – *feelings*, *rosa* – *dew*, *pink* in German), or they are (iii) very rare, again almost terminological concepts (*otava* – *aftermath* in agriculture, *plebejský* – *plebeian*, *řemdiš* – *flail*, a medieval weapon, *prvohory* – *paleozoic*).

6.3 Syntactic challenges

Czech allows flexible word order, whereas English and, to some extent, German follow more fixed patterns. This makes the translation of some exercises, especially gap-filling, challenging. If the words in the gap are omitted in the source sentence, a correct translation may be impossible even for human translators in some cases. Luckily, our data contain the correct option and several incorrect ones for each gap. However, we need to preserve the (complex) XML markup for each gap as well, which is also challenging, as shown below.

6.3.1 Word order in gap-filling

A naive approach that translates each XML element independently often fails because it cannot produce the correct target-language word order:

- (6) **SRC_{cs}**: *Z [vodní páry] vznikají oblaky.*
MT_{en}: From [water vapor] clouds are formed.
HT_{en}: Clouds are formed from [water vapor].

6.3.2 Compound alignment in gap-filling

Even if we translate the whole sentence without tags and then apply label projection using word alignment (see §2), the task is challenging because the alignment is not always one-to-one.

- (7) **SRC_{cs}**: *Severní Ameriku od Asie odděluje [Beringův] průliv.*
HT_{de}: Nordamerika ist durch die [Bering]straße von Asien getrennt.

The two words *Beringův průliv* (Bering Strait) should be translated as a single German compound word, *Beringstraße*. The web portal may not be ready to present gaps as part of a word (in our case, it renders a drop-down menu followed by a space). An alternative is to translate the word in the gap

with a final hyphen: *Bering-*, but this decision is based on the final presentation capabilities.

6.3.3 Non-monotonic alignment in gap-filling

In German, parts of verbs (such as infinitives in modal constructions or participles) tend to appear in sentence-final position (e.g., *odděluje* (*separates*) is translated as *ist ... getrennt* in Example 7). This can result in long-distance reordering and possibly in a different order of the gaps (e.g., if *odděluje* would be a gap). The XML has ID attributes attached to each gap, but the web portal may not be adapted to accommodate different orders of the gaps in each language.

6.3.4 Distractors in gap-filling

Originally, we applied a preprocessing method of gap-filling XML markup where the correct option is translated within the sentence, while the incorrect options (distractors) are translated separately afterwards. This improved the translation quality, both in terms of the form (inflection in Ukrainian and German) of the correct option as well as the rest of the sentence. However, the incorrect options were still often translated incorrectly due to the lack of context. In preliminary experiments, we tried to translate the incorrect options within the context of the whole sentence as well, but we noticed an increased number of mis-corrected counterfactual statements (see §6.4.1 below). Additionally, we have noticed that the rest of the sentence may be translated differently for each option and using the correct-option context may provide hints that make the exercise too easy, e.g. by revealing the grammatical gender of the correct option.

6.4 Other challenges

The following issues are either closely related to the translation method, like those in previous sections, or they are connected to the purpose of the translation (interaction with images in exercises).

6.4.1 Counterfactual statements

There are exercises with yes/no questions that contain counterfactual (i.e. false) statements. However, the LLM-based systems tend to correct false statements during translation:

- (8) **SRC_{cs}**: *Prvoka trypanozómu, který vyvolává spavou nemoc, přenáší octomilka obecná.*
MT_{en}: The trypanosome protozoan that causes sleeping sickness is transmitted by the *tsetse fly*.

						modrá
	11	12	13	14	15	16
	21	22	23	24	25	26
červená	31	32	33	34	35	36
	41	42	43	44	45	46
	51	52	53	54	55	56
	61	62	63	64	65	66

**Házíme červenou a modrou
hrací kostkou. Urči, jaká
je šance, že padne součet
alespoň 10.**

Figure 1: Example of embedded Czech text requiring in-image translation (*We roll a red die and a blue die. Determine the probability that the sum is at least 10.*)

HT_{en}: ... by the *fruit fly*.

This way, the originally false statement (*trypanosome is transmitted by the fruit fly*) is converted into a true statement (*trypanosome is transmitted by the tsetse fly*), which makes the whole translated yes/no exercise harmful for the students. Our plan is to instruct the LLMs in the system prompt that false statements should not be corrected and to provide the whole exercise for translation (including instructions such as “Are the following statements true?”).

6.4.2 Hallucinations

We encountered cases where the LLM-based translation completes an unfinished statement, which spoils the gap-filling exercise:

(9) **SRC_{cs}:** *Nad psaným textem většinou najdeme titulek neboli...*

MT_{de}: Über dem geschriebenen Text finden wir meist einen Titel oder eine Überschrift. (*lit. ‘Above a written text, we usually find a title or heading.’*)

HT_{de}: Über dem geschriebenen Text finden wir meist einen Titel oder...

This issue, similarly to the non-existent words (§6.2.3) and counterfactuals (§6.4.1), may be mitigated by adapting the LLM system prompt. However, we noticed that our SFT (and CPO/SimPO) fine-tuning decreases the model’s capability to follow general instructions.

6.4.3 In-Image Translation

Exercises are often accompanied by images. If an image is purely decorative, it can usually be ignored during translation. However, some images,

such as in Figure 1, contain embedded text (e.g., labels, captions, or instructions) that is necessary to solve the exercise. In such cases, the pipeline should apply in-image machine translation (MT) so that learners see the original visual content with the embedded text rendered in the target language.

In preliminary experiments, we examined whether low-parameter open-source Vision Language Models (VLMs) can detect images whose embedded text should be translated. On 1,000 images randomly sampled from the exercise image store, the best-performing model reached 84% recall ($F1 = 0.64$). Given that fewer than 7% of the sample actually required translation, this level of recall suggests the detector can surface most relevant images while filtering out the large majority that do not warrant resource-intensive in-image MT. See Appendix B for details.

For in-image MT, we plan to use more capable VLMs to extract the embedded text, translate it, and render the translation back into the image. Alternatively, one could generate a translated image in a single step, but this risks modifying non-textual content.

7 Conclusion

We presented a project that employs various MT methods for translating educational content – interactive web-based exercises – from Czech to Ukrainian, English, and German. We described several experiments to achieve the best translation quality using open source LLMs (EuroLLM-9B-Instruct), in particular the process of fine-tuning and further experiments on the given domain. We selected the three most promising systems using automatic MT metrics. Two rounds of manual evaluations helped to select the best system (SFT+CPO) and observe its performance in a real-world scenario within the web portal. We were able to significantly improve our original Pre-LLM system for Czech-Ukrainian in terms of both automatic (+1.5 chrF2, -0.4 MetricX24) and human (+0.61 score on a 0–10 scale) evaluation, thus confirming the superiority of LLMs with SFT+CPO adaptation for the education domain.

The second goal of the paper was to conduct an in-depth analysis of the translation errors, along with a description of the advantages and limitations of the approach chosen for our use case. Translating such a web portal required a substantial amount of engineering work as well as experimentation,

particularly in the area of terminology, word order and syntax in gap-filling exercises, and in-image translation. These issues have not yet been fully resolved, but we have outlined the next steps to address them. We have classified the translation errors linguistically: some types of lexical errors are the most significant issue and can occasionally impair the functionality of the entire exercise. At the same time, we tried to distinguish between traditional, well-documented MT errors and newly emerging ones, which are more typical of LLM-based translation methods. These errors include hallucinations, such as unintended corrections of false statements, or completing unfinished statements.

The translation systems were published in 2025 in the LINDAT/CLARIAH repository, the release consists of two items: a package of three machine translation models adapted for educational use: Czech-Ukrainian, Czech-English, and Czech-German (Hrabal et al., 2025)⁷ and a second package with an improved version of the translation software required for the newly released models (Popel et al., 2025)⁸. This package includes three tools: a web interface (Charles-translator-web-frontend) for machine translation with phonetic transcription of Ukrainian suitable for Czech speakers, an API server, and a tool for translating tagged documents including html, docx, odt, pptx, pdf, and odp. We will release updated system versions during 2026 in the same way. Furthermore, by the end of 2026, translated exercises will be made freely available to the public via Fraus-Klett educational web portal *Škola s nadhledem*.

Limitations

An obvious limitation is the disparity in capabilities between open-source and the newest commercial models. It can certainly be expected that all types of models will continue to improve quickly in tasks such as machine translation, and that current versions of large commercial models will perform better. On the other hand, data confidentiality is an issue. In our approach, all data remains on our servers and those of our partner company. Second, as far as we could tell, some translation issues remain unsolved in large commercial models, mainly concerning translations to or from small languages and highly specialized terminology (see 6.2.1).

⁷<http://hdl.handle.net/11234/1-6032>

⁸<http://hdl.handle.net/11234/1-6033>

As stated in §5, the evaluation served mainly as a quality-assurance and decision-making mechanism within the product development cycle, rather than as an attempt to establish annotator-independent ground truth or to measure reproducibility of theoretical judgments. For this reason, inter-annotator agreement was not treated as a central methodological requirement. In this applied evaluation context, practical effectiveness and deployment readiness took precedence over annotation-consistency metrics typical of hypothesis-driven research.

Ethical Considerations

This research was conducted in accordance with the ACM Code of Ethics and Professional Conduct. The work does not involve harmful applications, and human annotation was performed with informed consent and fair compensation. We aim to ensure transparency, reproducibility, and responsible release of resources.

Acknowledgments

This work was supported by the project TQ01000458 (EdUKate) financed by the Technology Agency of the Czech Republic (www.tacr.cz) within the Sigma 3 Programme. It has been using language resources and tools developed and/or stored and/or distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2023062).

We would like to thank our colleagues at Fraus-Klett Publishing: Adam Jelínek, Barbora Bartošová, Matěj Sutř, Barbora Tvarohová and Kateřina Berková for our ongoing cooperation.

References

- Nuaman Chen, Li Zhou, and Zilong Wang. 2023. Frustratingly easy label projection for cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4500–4515, Singapore. Association for Computational Linguistics.
- Miquel Esplà-Gomis. 2009. *Bitextor: a free/open-source software to harvest translation memories from multilingual websites*. In *Beyond Translation Memories: New Tools for Translators Workshop*, Ottawa, Canada.
- Miroslav Hrabal, Martin Popel, Lucie Poláková, Michal Novák, Věra Kloudová, and Mariia Anisimova. 2025. *EdUKate translation models 2025*.

- LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google Submission to the WMT 2024 Metrics Shared Task. In *Proceedings of the Workshop on Machine Translation (WMT 2024)*.
- Anna-Lena Kempe and Åke Grönlund. 2019. Collaborative digital textbooks—a comparison of five different designs shaping teaching and learning. *Education and Information Technologies*, 24(5):2909–2941.
- Věra Kloudová, Ondřej Bojar, and Martin Popel. 2021. [Detecting post-edited references and their effect on human evaluation](#). In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 114–119, Online. Association for Computational Linguistics.
- Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakouagna, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025. [Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413, Suzhou, China. Association for Computational Linguistics.
- Tom Kocmi, Martin Popel, and Ondřej Bojar. 2020. [Announcing CzEng 2.0 Parallel Corpus with over 2 Gigawords](#). *Preprint*, arXiv:2007.03006.
- Lieve Macken, Laura Van Brussel, and Joke Daems. 2019. NMT’s wonderland where people turn into rabbits. A study on the comprehensibility of newly invented words in NMT output. *Computational Linguistics in the Netherlands Journal*, 9:67–80.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Maslahatun Nadziro, Aufia Aisa, and Rina Dian Rahmawati. 2023. The development of QR code-based Arabic textbooks for non-arabic education students. *Scaffolding: Jurnal Pendidikan Islam dan Multikulturalisme*, 5(2):960–981.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Hefernan, and 1 others. 2022. [No language left behind: Scaling human-centered machine translation](#). *arXiv preprint arXiv:2207.04672*.
- Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2024. [Contextual label projection for cross-lingual structured prediction](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5738–5757, Mexico City, Mexico. Association for Computational Linguistics.
- Lucie Poláková, Martin Popel, Věra Kloudová, Michal Novák, Mariia Anisimova, and Jiří Balhar. 2025. Mitigating language barriers in education: Developing multilingual digital learning materials with machine translation. In *EDULEARN25 Proceedings*, pages 8754–8760, Valencia, Spain. IATED.
- Martin Popel, Michal Novák, Jiří Balhar, Ondřej Košarko, Jiří Mayer, Lucie Poláková, Věra Kloudová, and Mariia Anisimova. 2025. [EdUKate translation software 2](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Martin Popel, Lucie Poláková, Michal Novák, Jindřich Helcl, Jindřich Libovický, Pavel Straňák, Tomáš Krabáč, Jaroslava Hlaváčová, Mariia Anisimova, and Tereza Chláňová. 2024. [Charles Translator: A Machine Translation System between Ukrainian and Czech](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3038–3045, Torino, Italy. European Language Resources Association.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- D. K. Thennal, Chris Biemann, and Hans Ole Hatzel. 2026. [Just Use XML: Revisiting Joint Translation and Label Projection](#). *Preprint*, arXiv:2603.12021.
- 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.
- Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM-based machine translation. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria.
- Thomas Zenkel, Joern Wuebker, and John DeNero. 2021. [Automatic bilingual markup transfer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendices

A Additional Examples of Mistranslations by Type

A.1 Lexical challenges:

Terminology (6.2.1):

- (10) **SRC_{cs}**: *svor*
MT_{en}: *svor*
HT_{en}: mica schist
MT_{de}: Schlüsselwort
HT_{de}: der Glimmerschiefer
- (11) **SRC_{cs}**: *motýlice obecna*
MT_{en}: common hawker
HT_{en}: Beautiful demoiselle
MT_{de}: Großer Blaupfeil
HT_{de}: Blauflügel-Prachtlibelle
- (12) **SRC_{cs}**: *candát obecný*
MT_{en}: European pike
HT_{en}: zander
MT_{de}: der Europäische Wels
HT_{de}: der Zander

Homonymous expressions (6.2.2):

- (13) **SRC_{cs}**: *sloupek*
MT_{de}: die Säule (lit. 'pillar')
HT_{de}: die Kolumne (lit. 'journalistic column')
- (14) **SRC_{cs}**: *kůže*
MT_{de}: die Haut (lit. 'skin')
HT_{de}: das Leder (lit. 'leather')
- (15) **SRC_{cs}**: *pěvci*
MT_{de}: Sänger und Sängerinnen (lit. 'singers')
HT_{de}: Singvögel (lit. 'songbirds')

Non-existent words (6.2.3):

- (16) **SRC_{cs}**: *obkládat / dlaždičkovat*
MT_{de}: flieschen
HT_{de}: fliesen / mit Fliesen belegen (lit. 'to tile')
- (17) **SRC_{cs}**: *směsný komunální odpad*
MT_{de}: Mischmüllgemisch
HT_{de}: Restmüll (lit. 'residual waste')
- (18) **SRC_{cs}**: *Zakavkazsko*
MT_{de}: Zakavkazien
HT_{de}: Südkaukasus (lit. 'South Caucasus')
- (19) **SRC_{cs}**: *traktor*
MT_{de}: Tractor
HT_{de}: der Traktor (lit. 'tractor')

Mistranslations based on form similarity (6.2.4):

In Example 21, the incorrect translation appears to have been triggered by confusion with *opakování* (repetition), Example 22 perhaps by interference from Slovak *len* (only), Example 23 by similarity to *míchat* (to mix, to stir), and Example 24 perhaps by the Czech abbreviations for *sobota* and *neděle* (Samstag and Sonntag).

- (20) **SRC_{cs}**: *city*
MT_{en}: city
HT_{en}: feelings
MT_{de}: Stadt
HT_{de}: Gefühle
- (21) **SRC_{cs}**: *opak*
MT_{de}: Wiederholung (lit. 'repetition')
HT_{de}: Gegenteil (lit. 'opposite')
- (22) **SRC_{cs}**: *len*
MT_{de}: nur (lit. 'only')
HT_{de}: Flachs / Leinen (lit. 'flax / linen')
- (23) **SRC_{cs}**: *mícha*
MT_{de}: mix (lit. 'mix')
HT_{de}: Rückenmark (lit. 'spinal cord')
- (24) **SRC_{cs}**: *sob polární*
MT_{de}: Sonntag polare (lit. 'polar Sunday')
HT_{de}: Rentier (lit. 'reindeer')

A.2 Syntactic challenges (6.3):

- (25) **SRC_{cs}**: *Znečištění vzduchu [může] pocházet z lidské činnosti.*
MT_{en}: Air pollution [can] originating from human activity.
MT_{de}: Luftverschmutzung [kann] entstammen menschlicher Aktivität.
- (26) **SRC_{cs}**: *Společnými znaky strunatců je [páteř/struna] hřbetní, [trubicovitá/rozptýlená] nervová soustava a [otevřená/uzavřená] cévní soustava.*
MT_{en}: The common characteristics of chordates are [string/spine] dorsal, [scattered/tubular] nervous system and [open/closed] vascular system.
MT_{de}: Gemeinsame Merkmale der Chordatiere sind: [Wirbelsäule/die Saite] rückseitig, [röhrenförmig/zerstreut] das Nervensystem und [offen/geschlossen] das Gefäßsystem.

B Detecting Images for Translation

Some images used in our online exercises contain embedded text that should be translated. To avoid unnecessary computation for in-image MT, it is desirable to automatically detect and filter such images. In principle, this decision may depend on the surrounding exercise context; here, we adopt a simplified setting and classify each image in isolation, based solely on the type of text it contains.

The decision is also target-language dependent. For English, German, and Ukrainian, numbers, mathematical symbols, and variables typically do not require translation. However, because Ukrainian uses the Cyrillic script, measurement units written in Latin script should be translated.

We constructed a dataset for this experiment from our image store associated with the web exercises, which contains over 31k images. We randomly sampled 1,000 images and manually annotated (i) whether the image contains text and (ii) whether the text should be translated. Following the simplified setting, annotation was performed without access to the surrounding exercise context. We therefore labeled text as “requires translation” if it would be unintelligible to a target-language reader without translation, which is a superset of the text that is strictly necessary to solve the exercise. In our sample, 292 images contain text, and 68 contain text that requires translation.

We experimented with three quantized, relatively small open-source multimodal models: gemma3-12b-qat,⁹ qwen2.5-omni-7b-gptq,¹⁰ and llama4-scout-17b-16e-1q4.¹¹ The first two use dense feed-forward layers, whereas the latter uses a mixture-of-experts architecture, with 109B total parameters but 17B active parameters. We use the prompt shown in Figure 2 in a zero-shot manner.

Results in Table 5 show that the F1 score for detecting images with text that requires translation falls in a narrow range of 0.62–0.64 across all models. However, because this is a filtering step, recall is more important than precision. Therefore, among systems with similar F1 scores, we prefer the one with the highest recall, which is gemma3-12b-qat in our case. Precision remains

⁹https://huggingface.co/google/gemma-3-12b-it-qat-q4_0-gguf

¹⁰<https://huggingface.co/Qwen/Qwen2.5-Omni-7B-GPTQ-Int4>

¹¹<https://huggingface.co/unsloth/Llama-4-Scout-17B-16E-Instruct-GGUF>

System prompt:

You are a classification assistant. Analyze the provided image for text and determine if translation is required based on these rules:

1. No text: The image contains no readable characters.

2. No text to translate: Text is present, but it is NOT in Czech, or it consists solely of numbers, mathematical symbols, or variables.

3. Text to translate: Text is present in Czech, including units of measurement.

Constraint: You must respond with exactly one of these three labels: ["No text", "No text to translate", "Text to translate"]. Do not provide explanations or preamble.

User prompt:

Analyze this image.
<IMAGE>

Figure 2: Prompt template used for identifying images whose embedded text requires translation.

Model	P	R	F1
gemma3-12b-qat	0.52	0.84	0.64
qwen2.5-omni-7b-gptq	0.50	0.79	0.62
llama4-scout-17b-16e-1q4	0.56	0.74	0.63

Table 5: The quality of identification images containing a text that requires translation (support = 68).

reasonably high; that is, the detector filters out most images that either contain no text or contain embedded text that does not require translation.

Evaluating LLM Workflows for Generating Clinical Communication Assessment Items: A Comparative Study with Subject-Matter Experts

Christopher Runyon, Peter Baldwin, Ian Micir, Kevin Frome
Stephanie Mann, Saed Rezayi, Keelan Evanini, Victoria Yaneva

NBME

(crunyon, pbaldwin, imicir, kfrome,
semann, srezayidemne, kevanini, vyaneva)@nbme.org

Abstract

Generative AI is increasingly used to accelerate assessment content development, yet its effectiveness for generating content used in complex assessment tasks for knowledge-rich domains such as medical education is unclear. This study evaluates automated LLM-supported workflows for generating patient-centered communication assessment items that allow students to practice their communication skills. We compared two content generation approaches—constrained linear and exploratory branching—each implemented with and without anchoring in vetted multiple-choice questions (MCQs). Ten subject-matter experts (SMEs) evaluated 80 communication items across six quality dimensions using structured rubrics. The constrained linear approach yielded better ratings than exploratory branching approaches, particularly for medical accuracy and alignment with learning objectives and patient-centered behaviors. MCQ anchoring did not improve medical accuracy. Only a minority of items met all criteria without requiring revision, and no items were unanimously approved by all SMEs. These findings underscore the importance of workflow design in LLM-supported assessment content generation, the continued need for human oversight, and the current limitations of automated content generation in medical education.

1 Introduction

Recent advances in generative AI have made AI-assisted content development one of the fastest-growing educational applications of this technology, with numerous examples reported in both research and operational contexts (see Bouguettaya et al. (2025) for an overview). Owing to their ability to generate coherent and grammatically well-formed narratives, generative models have been used primarily in language-focused assessment domains such as language learning (Attali et al.,

2022; LaFlair et al., 2023) and reading comprehension (Lin and Chen, 2024; Säuberli and Clematide, 2024; Uto et al., 2023). In these settings, generative AI typically serves to accelerate item development even while extensive fairness and bias reviews and item performance analyses remain necessary to ensure that the resulting items function appropriately (Belzak et al., 2023).

In contrast, the application of generative AI to content development in knowledge-rich domains such as medicine, law, and engineering, among others, remains comparatively underexplored. These domains impose additional requirements beyond grammatical accuracy and narrative coherence. Generated content must accurately represent complex domain knowledge, including factual correctness (e.g., medical accuracy), currency (reflecting rapidly evolving knowledge bases), and domain plausibility (e.g., clinically realistic scenarios). Meeting these requirements poses a substantial generation challenge. Moreover, *evaluating* AI-generated content in knowledge-rich domains requires specialized expert judgment, making the design and execution of human evaluation studies complex and resource-intensive. As a result, robust methodological approaches for generating and validating assessment content in such domains require focused attention from the educational research community.

In this study, we investigate the use of generative AI (GPT5; OpenAI, 2025) to support content development in the domain of medical education. Specifically, we focus on developing and evaluating methods for generating clinical communication assessment items for the Communication Learning Assessment (CLA) tool. CLA is a mobile application designed to help learners practice clinical communication skills through interactions with virtual patients. The system presents learners with brief clinical scenarios and patient prompts, and learners record spoken responses that demon-

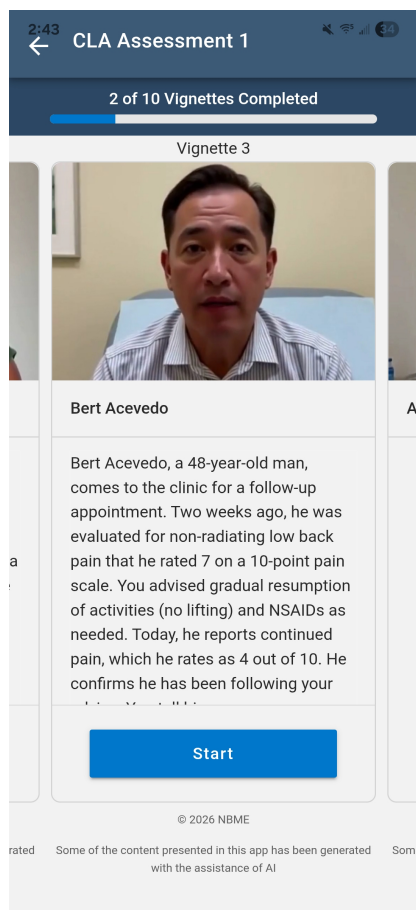


Figure 1: Screenshot of the Communication Learning Assessment tool.

strate their ability to use appropriate communication strategies (see Figure 1 for an illustration). The application then provides feedback to support skill development. For example, a learner may receive background information describing a clinical scenario such as a patient expressing concern over their weight-loss efforts plateauing (vignette background), followed by a patient statement or question (learner prompt). The learner is then asked to record a response that demonstrates their ability to use appropriate communication skills for that situation.

Developing suitable content for assessing clinical communication includes the generation and evaluation of multiple item components. To guide the reader's understanding of these, an example CLA item is shown in Appendix A.1. First, each item must include a medically-accurate and plausible clinical scenario appropriate for a target learner's level of training (e.g., first-year medical students). A given clinical scenario is called a *vignette*, as shown in Appendix A.1. The vignette must align with specific learning objectives (LOs)

for the assessment (e.g., “respond to emotion”), and be designed to elicit appropriate contextualized communication behaviors. These communication behaviors, also shown in the example, are referred to as *patient-centered behaviors* (PCBs; e.g., “provide empathic reassurance that her fear is understandable and manageable”). In addition, the learning experience concludes with a reflection on two exemplary “gold standard” responses (*exemplars*) that demonstrate how to effectively and appropriately communicate within the given scenario, each of which integrate all the targeted PCBs. Exemplars are a crucial aspect of content development that require alignment with the scenario and the targeted LOs and PCBs while maintaining a high degree of realism. In addition to medical accuracy, plausibility, and appropriate difficulty targeting, all components of the item—the vignette, LOs, PCBs, and exemplars—must be internally consistent and must not reinforce or perpetuate unsupported demographic associations or stereotypes (i.e., links not grounded in established clinical evidence). CLA also requires high-quality content covering a range of clinical communication scenarios. Together, these requirements make the development of high-quality CLA content challenging.

This study makes the following original contributions in “jump-starting” the development of clinical communication vignettes via AI:

- Propose and compare two fully automated content-generation methods that reflect different conceptual approaches to producing interdependent content: constrained linear versus exploratory branching.
- Evaluate whether grounding the generation process in a vetted multiple-choice question (MCQ) improves the medical accuracy of the output.
- Develop detailed evaluation rubrics and guidelines tailored for assessing AI-generated clinical communication items.
- Recruit and train ten subject-matter experts to evaluate 80 AI-generated items, each consisting of a vignette, LOs, PCBs, and exemplars.
- Analyze the evaluation results to assess the extent to which AI can reduce human effort in item development and identify areas for improving the generation process.
- Provide a detailed discussion of the contributions, limitations, and ethical considerations of using generative AI to assist in clinical con-

tent development.

Before we introduce details about the current study, Section 2 presents related work in AI-assisted content development in medicine for item formats beyond MCQs.

2 Background

2.1 Related Research

A robust medical education program teaches learners to do more than select a correct answer to an MCQ. Students must also learn to reason through complex diagnostic scenarios, such as interactions with standardized patients in objective structured clinical examinations (OSCEs). While LLMs have been used productively in MCQ item writing by supporting item development workflows (Kiyak and Emekli, 2024; Artsi et al., 2024), complex performance tasks pose a greater challenge. Yet, it is *because* these tasks are typically more resource-intensive to develop than their MCQ analogs that makes them natural—although more difficult—candidates for LLM-assisted content generation (Cusimano et al., 1994; Kiyak and Emekli, 2024; Miri et al., 2024).

Recent literature has begun to examine the potential for LLMs to assist with the generation of performance task materials, especially those materials related to OSCE and standardized-patient (SP) tasks. Misra and Suresh describe how ChatGPT can support OSCE preparation—including drafting cases, SP training materials, and grading rubrics—while emphasizing that medical accuracy, security, and compliance require systematic human validation (Misra and Suresh, 2024). Zafar et al. provide practical guidance for using AI to generate high-quality OSCE stations, recommending outcome alignment, structured prompts, iterative refinement, and faculty oversight to manage hallucinations and construct-validity threats (Zafar et al., 2026). Complementing peer-reviewed studies, the AAMC has released a workflow for generating curriculum-aligned, role-locked virtual patient profiles for simulation, explicitly positioning AI generation as scalable but educator-supervised (Quiroga Velasquez, 2026). Moving from static materials to interactive performance tasks, Brugge et al. show that LLM-simulated history-taking encounters paired with AI-generated structured feedback can improve learners’ clinical decision-making indicators, suggesting that LLMs may help generate and sustain practice-ready performance-assessment experiences when

embedded in governed workflows (Brügge et al., 2024). Prior work has explored using LLMs to support communication tasks, but these models have primarily served as the delivery mechanism for the task and its feedback (Stamer et al., 2023). In contrast, the use of LLMs to generate the content for communication tasks themselves remains underexplored.

3 Method

3.1 Generation Methods

The four generation methods are outlined below: 1a, 1b, 2a, and 2b. The core distinction between the “a” and “b” versions of each method is what anchors the generation at the start. In the “a” versions, the workflow begins with a diagnosis and communication task extracted from a retired clinical MCQ from a medical licensure examination. In the “b” versions, the workflow also includes the full text of the associated MCQ. Because these seed MCQs were previously developed in conjunction with physicians, the intent is that clinically accurate scenario details and logic can be inherited by the CLA item within the workflow. Each generation method developed a set of 20 items for a total of 80 items to be reviewed by subject-matter experts (SMEs). SMEs evaluated all four versions of each item, which were presented in randomized order.

3.1.1 Methods 1a/1b

Method 1 treats item writing as a progressive, linear build, similar to drafting a single coherent case from start to finish¹. It begins by establishing key guardrails for what the item should measure (the construct), either explicitly through encounter features defined up front (1a) or implicitly through the narrative constraints embedded in a retired MCQ stem (1b). The item is then generated in sequence, with each component checked for consistency with the selected learning objectives and intended tone/affect. Quality gates are applied throughout (e.g., accuracy and internal-consistency checks followed by revision as needed), and a final pass focuses on naturalness and readability. The build proceeds in the following steps, using the information from the preceding steps to generate subsequent content: identifying appropriate LOs based on the communication task and diagnosis → generating the communication vignette (including site of care and patient age) → generating the

¹Method 1 used gpt5-2024-12-01-preview

PCBs → generating the exemplars → stereotype check, selection of remaining demographics, and final refinement. The prompts used for Method 1a are available in the Appendix A.2.

3.1.2 Methods 2a/2b

Method 2 was designed as a deliberate alternative to Method 1 and emphasizes an exploratory branching search and refinement rather than early commitment to a single narrative². Instead of progressing along one path, it begins by generating multiple candidate cases using multiple agents (using the same LLM) and then converges through structured selection and improvement, such as ranking, justification, critique from different perspectives, and targeted revision. Early in the workflow, learning objective-mapped PCBs are used as a temporary scaffold while the vignette remains flexible, and the process relies on detailed component descriptions and repeated examples to guide both generation and critique. The workflow explores alternatives by branching across aspects of the vignette background before selecting and polishing the strongest candidate, and it delays introducing details such as site of care and patient demographics to reduce the chance that surface features steer construct specification too early. Overall, Method 2 treats alignment as something achieved through convergent optimization across the vignette, PCBs, and exemplars, rather than as a one-pass enforcement of early constraints.

3.2 Subject-Matter Expert Evaluation

Ten SMEs were recruited via NBME's Assessment Alliance network³ to provide feedback on the items. The SMEs were selected based on their involvement in the teaching of communication skills to medical students, a demonstrated research history in this area, experience in developing assessments in medical education, or a combination of these factors; nine were licensed medical doctors and the other was a PhD RDN with expertise in communication and sociocultural aspects of health care.

The SMEs evaluated each item based on the following item quality dimensions: *medical accuracy*, *LO alignment*, *PCB alignment*, *Exemplar alignment*, *presence of harmful stereotypes*, and *difficulty appropriateness* (see Appendix A.3 for a full definition of each dimension). These dimensions

were chosen because they reflected the most common areas of editorial work identified by the test development team responsible for communication item development.

For each dimension, SMEs were provided with a forced-choice Likert scale response (see Appendix A.4 for exact phrasing of the item-evaluation questions). For the first four item dimensions, the response options were about the degree of changes necessary for the accuracy / alignment, with options of “No Changes”, “Minimal”, “Moderate”, “Major”. Additional guiding language provided contextual information on response options. For stereotype inclusion, SMEs simply indicated if the vignette perpetuated any harmful stereotypes (Yes/No), and for difficulty appropriateness SMEs identified if the item was too easy, too difficult, or at the appropriate level for the student given the level of medical training targeted by the item.

It is important to note that in cases where an SME would judge a vignette to be medically inaccurate to the point of being not salvageable, they were permitted to opt out of providing ratings for subsequent item dimensions. This was a deliberate feature of the evaluation design rather than an incidental source of nonresponse. The intent was to avoid collecting downstream judgments for items that would not realistically proceed in an editorial workflow. For this reason, later-dimension analyses should be interpreted conditionally, that is, as characterizing items that satisfied the medical accuracy criterion.

3.3 Research Questions

- RQ1: How well does each generation method produce content that is aligned with each of the six item quality dimensions?
- RQ2: Does anchoring an item generation workflow with an MCQ improve the medical accuracy of the generated items?
- RQ3: How many items require no changes across all item dimensions, and thus could be directly used without human refinement?

3.4 Statistical Analysis

SME ratings of AI-generated content were analyzed using cumulative link mixed models (CLMMs; Christensen, 2019), an extension of ordinal logistic regression that accommodates clustered measures designs. CLMMs are appropriate when outcomes are ordinal categories with a natural ordering but spacing between levels is not assumed

²Method 2 used gpt5-2025-01-01-preview.

³<https://www.nbme.org/research/research-collaborations/assessment-alliance>

to be consistent; that is, the difference between "None" and "Minimal" is not assumed to be the same as the difference between "Moderate" and "Major".

CLMMs model the cumulative probability of observing a response at or below each category. The model estimates threshold parameters that define boundaries between adjacent categories on a latent continuous scale, allowing the data to determine category spacing rather than imposing assumptions. CLMMs extend standard ordinal regression by incorporating random effects, which account for systematic variation due to nested observations. In our design, all 10 raters evaluated all 80 case-method combinations (a fully-crossed design), introducing dependencies that would violate independence assumptions of standard regression. The CLMM addresses this by estimating random intercepts for raters and cases, capturing individual rater tendencies (e.g., overall harshness or leniency) and case-level variation in quality.

CLMMs also partition variance into components attributable to random effects (raters and cases) and residual variance, providing insight into sources of rating variability. Variance decomposition follows standard practice for logistic mixed models (Hedeker and Gibbons, 2006); residual variance is fixed at $\pi^2/3$ and random effect variances are estimated via maximum likelihood. The proportion of variance attributable to each random effect is computed as its estimated variance divided by the total (rater variance + case variance + $\pi^2/3$). The proportion of variance attributable to raters reflects consistency in how raters applied evaluation criteria: lower rater variance indicates that raters evaluated content similarly, while higher rater variance suggests systematic differences in stringency or leniency across raters. The proportion attributable to cases reflects genuine variability in content quality across the 20 diagnosis-communication task pairs common to all generation methods. Together, these variance components offer a complementary perspective on rating consistency alongside the fixed effects of primary interest.

Model coefficients are expressed as log-odds, with exponentiated values interpretable as odds ratios. An odds ratio greater than 1 indicates higher odds of receiving a more severe rating relative to the reference method; values less than 1 indicate lower odds. The proportional odds assumption, required for CLMMs, states that predictor effects (generation methods) are consistent across category

thresholds; this assumption was evaluated using likelihood ratio tests. Models were fit using the ordinal package in R with Laplace approximation for random effects integration (Christensen, 2019).

The cumulative link mixed model (Christensen, 2018) is specified as:

$$\text{logit}[P(Y_{ijk} \leq c)] = \theta_c - (\beta_k + u_i + v_j) \quad (1)$$

where Y_{ijk} is the ordinal rating from rater i for case j under generation method k ; θ_c are threshold parameters separating adjacent response categories, constrained such that $\theta_1 < \theta_2 < \dots < \theta_{C-1}$; β_k is the fixed effect for method k (with one method serving as reference); and $u_i \sim N(0, \sigma_{\text{rater}}^2)$ and $v_j \sim N(0, \sigma_{\text{case}}^2)$ are crossed random intercepts for raters and cases, respectively.

4 Results

4.1 Medical Accuracy

Table 1: Medical Accuracy

Method	OR (95% CI)	p
Method 1b	1.28 (0.80, 2.04)	0.29
Method 2a	2.18 (1.39, 3.42)	< 0.001
Method 2b	2.98 (1.90, 4.67)	< 0.001

We fit a cumulative link mixed model estimating the generation method as a fixed effect and random intercepts for raters and cases. Method 1a served as the reference for all analyses. Table 1 shows that items generated by Methods 2a and 2b were significantly more likely to receive ratings indicating they required additional changes for medical accuracy. Method 1b did not significantly differ from Method 1a. The proportional odds assumption was supported ($\chi^2(6) = 5.99, p = .42$). Descriptive statistics for expert ratings for each generation method across all item dimensions are reported in Appendix A.5.

About one-third of the variance in ratings (33.1%) was due to systematic differences between raters, indicating notable variability in rater stringency. Case-level differences accounted for only 8.3% of variance, suggesting that content quality was relatively consistent across cases. The majority of variance (58.6%) was residual, reflecting rating-level variability not captured by rater or case effects.

4.1.1 Effect of Using Full MCQs on Medical Accuracy

Table 2: Effect of MCQ on Medical Accuracy

IV	OR (95% CI)	p
Method 2	2.18 (1.39, 3.42)	< 0.001
"b" Methods	1.28 (0.81, 2.04)	0.29
Interaction	1.07 (0.57, 1.98)	0.84

To isolate the effect of incorporating MCQs into the generation workflow (RQ2), a separate CLMM was estimated with generation approach (Method 1 vs. Method 2), starting point (diagnosis and communication task only—the "a" methods—vs. full MCQ item content—the "b" methods), and their interaction included as fixed effects. Method 1 and the "a" generation methods were the reference groups. For the main effect of method, Method 2 produced content requiring significantly more revision than Method 1 (Table 2). The main effect for the inclusion of the full MCQ item content did not significantly affect medical accuracy ratings, and no interaction was observed, indicating that the advantage of Method 1 over Method 2 for medical accuracy was consistent regardless of generation workflow starting point. Variance decomposition indicated that rater tendencies (33.1%) exceeded case differences (8.3%), with 58.6% residual variance. The proportional odds assumption was supported for both generation method ($\chi^2(1) = 0.06, p = .80$) and starting point ($\chi^2(1) = 0.07, p = .80$).

4.2 Learning Objective Alignment

Table 3: Learning Objective Alignment

Method	OR (95% CI)	p
Method 1b	1.14 (0.45, 2.84)	0.78
Method 2a	14.25 (6.51, 31.18)	< 0.001
Method 2b	4.73 (2.12, 10.57)	< 0.001

For learning objective alignment (Table 3), generation Methods 2a and 2b showed significantly poorer alignment than Method 1a. Method 1b did not differ significantly from Method 1a. Variance decomposition indicated that rater tendencies (22.9%) and case differences (23.5%) contributed approximately equally to random variance, with 53.6% residual variance. The proportional odds assumption was supported ($\chi^2(6) = 8.01, p = .23$).

4.3 Patient-Centered Behavior Alignment

Table 4: Patient-Centered Behavior Alignment

Method	OR (95% CI)	p
Method 1b	0.56 (0.30, 1.04)	0.07
Method 2a	5.57 (3.29, 9.41)	< 0.001
Method 2b	3.56 (2.09, 6.06)	< 0.001

Here, the proportional odds assumption showed evidence of violation ($\chi^2(3) = 9.43, p = .02$). Given the modest departure and to maintain consistency with analyses of other outcomes, we retained the proportional odds model; however, results should be interpreted with appropriate caution.

For PCB alignment, Table 4 shows that content from Method 2a and Method 2b received significantly worse ratings than Method 1a content. Method 1b showed a trend toward better performance than Method 1a, though this was not statistically significant. Variance decomposition indicated that case differences (16.5%) exceeded rater tendencies (8.5%), with 75% residual variance, suggesting that raters applied criteria consistently while cases varied meaningfully in quality. As a reminder, the proportional odds assumption was violated and readers are cautioned against over interpreting these findings.

4.4 Exemplar Alignment

Table 5: Exemplar Alignment

Method	OR (95% CI)	p
Method 1b	1.36 (0.90, 2.07)	0.14
Method 2a	0.81 (0.51, 1.25)	0.34
Method 2b	0.74 (0.47, 1.18)	0.21

For exemplar alignment, no significant differences were observed between development methods (all $p \geq .14$). Variance decomposition revealed that case differences accounted for only 0.9% of variance, suggesting that content was uniformly aligned with exemplars regardless of case or method. Rater tendencies contributed 16.3% of variance, with residual variance comprising 82.8%. The proportional odds assumption was again supported ($\chi^2(6) = 9.38, p = .15$).

4.5 No Stereotypes

Stereotype inclusion was rare, flagged in only 13 of 723 ratings (1.80%) across all methods. Due to this low base rate, formal inferential comparisons were not conducted. Descriptively, rates were similarly low across methods (Method 1a: 2.7%;

Method 1b: 2.7%; Method 2a: 1.1%; Method 2b: 0.6%). Although stereotype flags were too rare to permit formal comparisons across methods, the potential consequences of stereotype inclusion are serious, underscoring the importance of maintaining vigilance throughout both item generation and subsequent human review.

Notably, the study design may have contributed to some stereotype flags. Because experts evaluated four versions of each case with potentially similar demographic characteristic combinations (race, gender, diagnosis), this may have prompted some raters to perceive stereotyping that reflected accumulated pattern recognition rather than content within any single version.

4.6 Appropriate Difficulty

Table 6: Appropriate Difficulty

Gen Method	OR (95% CI)	<i>p</i>
Method 1b	0.86 (0.48, 1.54)	0.62
Method 2a	0.66 (0.37, 1.16)	0.14
Method 2b	0.52 (0.29, 0.92)	0.02

Response options for difficulty were "too easy," "appropriate," and "too difficult," providing granular insight into the generation process. However, our primary interest was whether each method produced items at an appropriate difficulty level. We therefore collapsed responses into a binary outcome: appropriate versus off-level (combining "too easy" and "too difficult"). Across all generation methods, the majority of items were judged to be at an appropriate difficulty level (80.8%). When items were off-level, they were more commonly rated as too difficult (17.5%) than too easy (1.7%).

A generalized linear mixed model with binomial family was fit with random intercepts for raters and cases (Table 6). Method 2b produced content significantly less likely to be rated at the appropriate difficulty level compared to Method 1a; Method 1b and Method 2a did not differ significantly from Method 1a. Variance decomposition indicated that rater tendencies (16.1%) exceeded case differences (8.3%), with 75.6% residual variance. Model assumptions were evaluated and met: no overdispersion (ratio = 0.83), random effects showed no significant departures from normality, and residual distributions were acceptable.

4.7 Full Item Evaluation

Table 7: All Alignment

Gen Method	OR (95% CI)	<i>p</i>
Method 1b	0.63 (0.40, 1.01)	0.06
Method 2a	0.40 (0.25, 0.66)	<0.001
Method 2b	0.38 (0.23, 0.52)	<0.001

To evaluate the total item quality (RQ3), a composite variable was created for each item / rater combination that indicated if a particular rater felt that item needed no corrections across all 6 dimensions (medically accurate; LO, PCB, and exemplar all aligned; no stereotype; appropriate difficulty). Of the 800 item/generation method/rater combinations, only 222 (27.8%) met this criterion. A generalized linear mixed model with binomial family (outcome was binary: all aligned vs. not) was fit with random intercepts for raters and cases. Table 7 indicates that Method 2a and Method 2b had significantly lower odds of meeting all criteria compared with Method 1a. Method 1b also demonstrated a lower success rate; however, this difference was not statistically significant. Variance decomposition indicated that rater tendencies (27.9%) exceeded case differences (4.7%), with 67.4% residual variance. Model assumptions were again evaluated and met: no overdispersion (ratio = 0.94), random effects showed no significant departures from normality, and residual distributions were acceptable.

This analysis does not capture the extent of rater agreement that all item dimensions were aligned within a given item–generation method pair. To assess this directly, we summed the number of the 10 SMEs who agreed that a given item required no changes for each of the 80 item–method combinations. Table 8 shows the distribution of rater agreement across the 80 item–method combinations. For example, 11 item–method combinations received no full alignment ratings from any rater, while only 1 combination was rated as fully aligned by 7 raters—the maximum observed; no more than 7 raters agreed that an item–method combination required no edits. An average of 2.78 raters agreed all criteria were met for any item–method combination. Figure 2 in the Appendix shows the count of raters indicating full alignment by case and generation method.

Table 8: Distribution of Items by Number of Raters Indicating Full Alignment

Raters in Full Alignment	Count	Percent (%)
0	11	13.75
1	15	18.75
2	13	16.25
3	10	12.5
4	12	15
5	12	15
6	6	7.5
7	1	1.25
8	0	0
9	0	0
10	0	0

5 Discussion

This study examined the viability of fully automated generative AI workflows for producing clinical communication items suitable for formative assessment in medical education. Across three research questions, the findings provide converging evidence that generation method matters substantially for item quality, that anchoring generation in vetted multiple-choice questions does not meaningfully improve medical accuracy, and that only a modest proportion of generated items are currently suitable for direct use without human refinement, suggesting human review will remain an important aspect of AI-assisted content generation in the medical communication domain.

The results for RQ1 show consistent differences between generation methods in their ability to produce content aligned with the six item quality dimensions. Constrained linear methods (Method 1a and 1b) outperformed exploratory branching methods (Method 2a and 2b) across several core dimensions, most notably medical accuracy, learning objective alignment, and patient-centered behavior alignment. These dimensions are foundational to the validity of clinical communication assessments, as they ensure that scenarios are both medically plausible and capable of eliciting the intended communication behaviors.

In contrast, exemplar alignment showed no significant differences across methods. This finding suggests that once a successful vignette has been generated, LLMs are generally capable of producing exemplar responses that require very little revision, if any.

Stereotype flags were uncommon across all workflows. Item difficulty targeting was generally insensitive to generation method; however, Method 2b items were more likely to be judged as

falling outside the intended difficulty range, most often being too difficult.

Collectively, these results suggest that while certain classes of failure were less likely to arise, medical accuracy and alignment among items' component parts remain highly sensitive to the structure of the generation workflow.

RQ2 addressed whether anchoring generative workflows in vetted multiple-choice questions improves the medical accuracy of the generated clinical communication items. Contrary to expectations, no evidence was found that incorporating full MCQs as starting points improved medical accuracy. This null effect held across both linear and branching generation methods, and no interaction between anchoring and method was observed.

These findings suggest that the presumed benefits of grounding generation in validated assessment content may not readily transfer from selected-response contexts to more open-ended performance tasks. Although MCQs encode medically correct information, their structure and purpose differ substantially from those of clinical communication items. Expanding an MCQ into a fully developed communication item requires substantial inference and contextual detail that the MCQ format does not provide or constrain. As a result, anchoring on MCQs may offer less practical value than anticipated for supporting accuracy in complex generative tasks.

The third research question (RQ3) focused on the practical yield of the generation workflows: how many items met *all* quality criteria without requiring human revision. Across methods, only a small minority of items satisfied this stringent standard, and items produced by branching generation workflows were significantly less likely to do so than those produced by linear workflows. Importantly, the number of raters agreeing that an item-method combination required no changes varied widely and there were no items for which all raters (or even 8/10 raters) agreed that all elements were aligned. These results point to the persistent limitations of fully automated generation for high-stakes or high-fidelity assessment content in medicine, while also revealing meaningful efficiency differences across workflows.

Importantly, the composite outcome integrates judgments across all six quality dimensions, reflecting the holistic standard that assessment developers must meet in practice. That some items—particularly from constrained linear work-

flows—were judged acceptable without revision indicates that AI-assisted generation can reduce human development effort, even if it does not eliminate the need for expert review.

Overall, these findings suggest that generative AI can play a productive but circumscribed role in the development of clinical communication assessment content. Workflow design emerges as a central determinant of quality, with simpler, more structured generation approaches outperforming more complex branching strategies. The results further caution against assuming that validated content from one assessment format can be straightforwardly repurposed to support generation in another.

From a broader perspective, this study illustrates the importance of evaluating AI-generated assessment content not only for surface-level plausibility but also for deep alignment among medical context, learning objectives, and elicited behaviors, when developing complex performance tasks. As generative models continue to improve, careful attention to workflow structure and evaluation criteria will remain essential for ensuring that efficiency gains do not come at the expense of validity.

6 Limitations and Ethical Considerations

Several limitations deserve comment. First, the evaluation relied on SME judgment across six dimensions of item quality. Although SMEs completed an orientation session and practice review before formal rating began, a single training session may not have been sufficient to ensure shared interpretations of all rubric categories. The nontrivial rater-level variability observed in several analyses may, in some cases, reflect construct-relevant differences in professional judgment among SMEs; it may also reflect differences in rating severity or rubric interpretation. Interestingly, the medical accuracy dimension showed the most variance across raters, although this dimension is arguably the most objective of all dimensions. Future work would benefit from additional training and rater calibration before the formal review task begins, both to refine rubric wording and to ensure a stronger shared understanding of the evaluation criteria.

Relatedly, we report variance decomposition rather than relying exclusively on traditional interrater agreement statistics because our data structure made single-summary agreement coefficients poorly aligned with our inferential goals. The combination of multi-point ordinal scales, a large rater

panel, and ratings concentrated in modal categories created conditions under which chance-corrected agreement statistics can be difficult to interpret. Kappa-family statistics, including Fleiss's kappa, are sensitive to prevalence and marginal distributions and can yield low estimates even when observed agreement is high (Feinstein and Cicchetti, 1990). Krippendorff's alpha is more flexible with respect to number of raters and scale type, but it also summarizes reliability as observed disagreement relative to expected disagreement derived from the overall distribution of ratings, which can be difficult to interpret when rating distributions are highly restricted or skewed (Hayes and Krippendorff, 2007). Gwet's AC2 is less sensitive to prevalence effects than kappa-based statistics (Gwet, 2002), but none of these coefficients separate systematic rater severity differences from residual disagreement about item quality. Variance decomposition better matched our goals because it partitioned variation into method, item, rater, and residual components, allowing us to evaluate method differences while explicitly modeling rater variability. This approach allowed us to examine whether raters exhibited consistent relative patterns across methods despite differences in absolute rating standards. Future work could develop or evaluate agreement indices better suited to ordinal, multi-rater designs with skewed response distributions and systematic rater severity effects.

Second, the review process followed a planned conditional evaluation design. SMEs were permitted to stop rating later dimensions when an item was judged medically inaccurate to the point of being unsalvageable, on the rationale that downstream judgments are of limited practical value for content unlikely to advance in an editorial workflow. This approach increases the practical relevance of the evaluation by aligning it with how content would be screened in operational development, but it also changes the scope of inference for later outcomes. Certain constructs—such as vignette alignment—are not meaningfully defined for medically unsalvageable cases, as alignment is a property of viable cases only. Accordingly, later evaluations are best interpreted not as judgments conditional on medical accuracy, but rather as judgments about medically accurate items.

Third, despite clear workflow differences across several outcomes, each method produced only a modest number of items that required no revisions. Therefore, the absence of a measurable benefit from

anchoring generation in retired MCQs should be interpreted as a lack of evidence for improvement under the present design rather than as definitive evidence that anchoring cannot be useful. The usefulness of anchoring likely depends in part on the source items selected: some MCQs may provide a richer substrate for communication-focused vignette generation than others, depending on the amount of clinically relevant context available and the extent to which that content helps the LLM perform the generation task. Future research could consider broader and more systematic sampling of MCQ items. Likewise, statistically significant differences across workflows should not be taken to imply that the absolute quality of generated items was high or that the observed advantages would necessarily translate into large practical gains in operational item development.

Fourth, the present comparison is bounded by the specific model and implementation used in the study. All items were generated using a single LLM (GPT5) and a fixed set of workflow designs. The findings should therefore not be generalized to all models, prompting strategies, or medical education contexts. The observed advantage of the constrained linear workflow over the exploratory branching workflow may reflect a broader principle—that more tightly constrained generation is beneficial when item components must remain strongly aligned—but it may also depend on particulars of the prompts, revision logic, and model behavior used here. More generally, differences across methods may reflect implementation details as much as any intrinsic advantage of one workflow family over another. Replication across models and alternative implementations of the same workflow concepts will be necessary to determine which conclusions are robust and which are due to specific characteristics of a given implementation. For this, it may be beneficial to include factors beyond content quality that nevertheless have practical importance such as the relative costs and run times across methods. In addition, the study relied on private LLM deployments that did not log data or use prompts for model training. Such secure deployment options may not be available in all settings, which could limit the feasibility of similar workflows when safeguarding assessment content is essential.

These limitations also point to several ethical considerations. Most importantly, the fluency of generated clinical content should not be mistaken

for evidence that the content is adequate for a particular assessment purpose. Validity evidence remains necessary to support the intended interpretations and uses of any assessment. Fairness considerations are likewise central. Although stereotype flags were too rare to support formal comparisons across methods, even infrequent instances can be consequential, particularly in educational materials involving patient demographics, diagnoses, and communication challenges. Ethical oversight should therefore extend beyond item-level review to consideration of aggregate patterns in representation across generated content pools. As a whole, these considerations suggest that the most defensible near-term role for generative AI in this context is to support early drafting while preserving expert review, fairness screening, and institutional control over source materials and final content decisions.

References

- Yaara Artsi, Vera Sorin, Eli Konen, Benjamin S. Glicksberg, Girish Nadkarni, and Eyal Klang. 2024. [Large language models for generating medical examinations: systematic review](#). *BMC Medical Education*, 24:354.
- Yigal Attali, Andrew Runge, Geoffrey T LaFlair, Kevin Yancey, Sarah Goodwin, Yena Park, and Alina A Von Davier. 2022. The interactive reading task: Transformer-based automatic item generation. *Frontiers in Artificial Intelligence*, 5:903077.
- William CM Belzak, Ben Naismith, and Jill Burstein. 2023. Ensuring fairness of human-and ai-generated test items. In *International conference on artificial intelligence in education*, pages 701–707. Springer.
- Sirine Bouguettaya, Francesco Pupo, Min Chen, and Giancarlo Fortino. 2025. [A meta-survey of generative ai in education: Trends, challenges, and research directions](#). *Big Data and Cognitive Computing*, 9(9).
- Emilia Brügge, Sarah Ricchizzi, Malin Arenbeck, Marius Niklas Keller, Lina Schur, Walter Stummer, Markus Holling, Max Hao Lu, and Dogus Darici. 2024. Large language models improve clinical decision making of medical students through patient simulation and structured feedback: a randomized controlled trial. *BMC medical education*, 24(1):1391.
- Rune Haubo B Christensen. 2018. Cumulative link models for ordinal regression with the r package ordinal. *Submitted in J. Stat. Software*, 35:1–46.
- Rune Haubo Bojesen Christensen. 2019. [ordinal: Regression Models for Ordinal Data](#). R package version 2019.12-10.

- Michael D. Cusimano, R. Cohen, W. Tucker, J. Mur-naghan, R. Kodama, and R. Reznick. 1994. [A comparative analysis of the costs of administration of an OSCE \(objective structured clinical examination\)](#). *Academic Medicine*, 69(7):571–576.
- Alvan R. Feinstein and Domenic V. Cicchetti. 1990. [High agreement but low kappa: I. the problems of two paradoxes](#). *Journal of Clinical Epidemiology*, 43(6):543–549.
- Kilem L. Gwet. 2002. Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity. *Statistical Methods for Inter-Rater Reliability Assessment*, 2:1–9.
- Andrew F. Hayes and Klaus Krippendorff. 2007. [Answering the call for a standard reliability measure for coding data](#). *Communication Methods and Measures*, 1(1):77–89.
- Donald Hedeker and Robert D. Gibbons. 2006. *Longitudinal Data Analysis*. Wiley, Hoboken, NJ.
- Yavuz Selim Kıyak and Emre Emekli. 2024. [ChatGPT prompts for generating multiple-choice questions in medical education and evidence on their validity: a literature review](#). *Postgraduate Medical Journal*, 100(1189):858–865.
- Geoff LaFlair, Kevin Yancey, Burr Settles, and Alina A von Davier. 2023. Computational psychometrics for digital-first assessments: a blend of ml and psychometrics for item generation and scoring. In *Advancing natural language processing in educational assessment*, pages 107–123. Routledge.
- Zhiqing Lin and Huilin Chen. 2024. Investigating the capability of chatgpt for generating multiple-choice reading comprehension items. *System*, 123:103344.
- Kheizaran Miri, Tahere Sarbooz Hoseinabadi, Ali Yaghobi, Sadaf Kholosi, and Mehdi Miri. 2024. [Cost management analysis of objective structured clinical examination \(OSCE\): guide to the universities of medical sciences](#). *BMC Medical Education*, 24:1241.
- Sanghamitra M. Misra and Srinivasan Suresh. 2024. [Artificial intelligence and objective structured clinical examinations: Using ChatGPT to revolutionize clinical skills assessment in medical education](#). *Journal of Medical Education and Curricular Development*, 11:1–6.
- OpenAI. 2025. ChatGPT-5. <https://chat.openai.com/>. Large language model.
- Adriana Quiroga Velasquez. 2026. [Medical education case generator GPT workflow](#). Association of American Medical Colleges (AAMC), Advancing AI Resource Collection. Last updated February 2026.
- Andreas Säuberli and Simon Clematide. 2024. Automatic generation and evaluation of reading comprehension test items with large language models. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REading Difficulties (READI)@ LREC-COLING 2024*, pages 22–37.
- Tjorven Stamer, Jost Steinhäuser, and Kristina Flägel. 2023. Artificial intelligence supporting the training of communication skills in the education of health care professions: scoping review. *Journal of Medical Internet Research*, 25:e43311.
- 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.
- Imran Zafar, Hadeel Aboueisha, Ibrahim Elhassan, Fouzia Shersad, Suleyman Ayhan Caliskan, Asma Fatima Syeda, Mohammed Al-Houqani, and Mohi Eldin Magzoub. 2026. [Ten tips for utilizing AI to generate high quality OSCE stations in medical education](#). *Frontiers in Medicine*, 12:1744657.

A Appendix

A.1 Generated CLA Example Item

Field	Generated Content
Gender	cisgender woman assigned female sex at birth
Race	WHITE White or Caucasian
Affect	fear (fear, anxiety, insecurity, inferiority, panic)
SiteOfCare	Emergency: Urgent treatment center
Diagnosis	asthma
Ctask	Benefits and risks of a medication
Vignette	Megan Porter, a 35-year-old woman, comes to the urgent treatment center with a 1-week history of chest tightness and nighttime dry cough. Over the past 3 months, she had two emergency visits for acute shortness of breath treated with bronchodilators. She reports frequent upper respiratory infections, takes no medications, and has a 10-year one-pack-per-day smoking history. After confirming normal vital signs, a clear chest x-ray, and bilateral end-expiratory wheezes, you explain that the findings are consistent with asthma, advise smoking cessation, and recommend a daily controller inhaler to reduce airway inflammation, outlining expected benefits and potential side effects. She fidgets, appearing anxious, and asks,
Prompt	"I'm nervous about starting this inhaler. I smoke; how will the inhaler help breathing at night and how can you help me avoid side effects?"
CaseExemplar1	I hear this makes you anxious. A daily controller inhaler, a preventer medicine, reduces swelling in your airways. It helps calm nighttime cough and tightness. It also lowers your risk of flare-ups, meaning asthma attacks. It can still improve your breathing at night even if you smoke. To keep side effects low, we will check your inhaler technique. You will use a spacer, a tube that helps more medicine reach your lungs, and rinse your mouth after each dose. We will start with the lowest dose that keeps you well, and follow up to adjust. It is okay to feel nervous. Your reaction makes sense, and we can manage this safely together so you feel comfortable starting. What worries you most about starting the inhaler?
CaseExemplar2	Thanks for telling me that. Starting a new inhaler can feel unsettling. A daily controller inhaler calms airway swelling, so it should ease your nighttime chest tightness and dry cough, and lower your risk of flare-ups. Even if you smoke, it should still help your breathing. We can keep side effects low by going over your technique together, using a spacer (a small attachable tube), rinsing your mouth after each use, starting at the lowest dose that works, and checking in to adjust based on how you feel. We can take it step by step. What worries you most about starting this inhaler?
LO1	Solicit the patient's perspective

Field	Generated Content
PCB1	Ask what worries her most about starting the inhaler.
LO2	Accurately explain the nature of the problem and/or its management
PCB2	Explain that daily controller medicine reduces airway swelling, nighttime symptoms, and flare-ups.
LO3	Address patient emotion
PCB3	Provide empathic reassurance that her fear is understandable and manageable.
SME comments about medical accuracy	"Might be nice to confirm her vitals as stable. It's unclear how sick she is."
SME comments about learning objective alignment	"The vignette and prompt alludes to addressing how to minimize side effects. This is not clearly addressed in LO/PCB. I actually think this aspect of the case may be too complex anyway so would modify the prompt and exemplar from asking this for this level of learner." "It would be more appropriate for a student to practice smoking cessation counseling in this scenario than teach about inhaler use, which is beyond their scope of knowledge."
SME comments about pcb alignment	None.
SME comments about exemplar alignment	"Might be a bit much to expect a student at this level to know how to minimize side effects." "Ex 2 better than Ex 1, especially with the mention of being anxious which feels a touch judgmental." "Exemplar 2 could provide more direct empathic reassurance to the patient."
SME comments about appropriate difficulty	"Not sure— I think knowing how to minimize side effects could be too advanced." "Would not expect the student to know how to counsel on inhaler use, but other than that it's fine." "I think this one finally gets it all correct!"

Note. LO = Learning Objective; PCB = Patient-Centered Behavior; SME = Subject Matter Expert.

A.2 Prompts for Workflow 1a

Initial Prompt: You are an expert clinical communication content designer. You are building content for a communication tool that provides medical professionals the opportunity to practice their communication skills. The content build is occurring in a step-by-step process, and this is just one step in the process. Do not generate any other case content beyond what is requested here. Do not modify, expand, or infer additional conditions, as those will be taken care of in subsequent steps.

I will provide you with a patient diagnosis and a communication task. The communication task is the purpose of the physician-patient communication (e.g., explaining the purpose of a vaccination, information about a low-risk procedure or medication, speak with the patient about a common or easily treatable diagnosis).

In this step you will review the diagnosis and communication task, and your task is to review this information and generate 3 fields: - AGE : The plausible age range of a patient with this particular diagnosis that is appropriate for the communication task, drawn from a lower bound of 18 years old and upper bound of 90 years old. The age range you should return should reflect the most common ages where this diagnosis and communication task might occur. - GENDER : The possible sex of the patient, either MAN, WOMAN, or EITHER. For this exercise we can assume the patient is cisgendered. - Site of Care : The possible locations where this communication task is taking place. Choose only those locations that are most common given the diagnosis and the communication task. The possible options are [Specific Case Guidelines]. Choose all sites of care that are applicable.

The structure of your output should be as follows: DIAGNOSIS: the exact diagnosis provided CTASK: the communication task as provided AGE: The plausible age range of the patient GENDER: The plausible gender(s) for the patient SOC: The plausible site(s) of care where the communication may be taking place Here are the inputs you need: DIAGNOSIS: {diag} CTASK: {comtask}

LO prompt (Step 2): "You are an expert clinical communication content designer. You are building content for a communication tool that provides medical professionals the opportunity to practice their communication skills. The content build is occurring in a step-by-step process, and this is just

one step in the process. Do not generate any other case content beyond what is requested here. Do not modify, expand, or infer additional conditions, as those will be taken care of in subsequent steps. I will provide you with a patient diagnosis and a communication task. The communication task is the purpose of the physician-patient communication (e.g., explaining the purpose of a vaccination, information about a low-risk procedure or medication, speak with the patient about a common or easily treatable diagnosis). I will also provide you with: a plausible age range (AGE) for the patient; the plausible gender for the patient (GENDER); and the plausible site(s) of care where the communication task is taking place (SOC). Here is that base case information: {step1output}

You have 2 tasks: *TASK 1* Identify the learning objectives that would most naturally occur in the communication scenario given the diagnosis and plausible age, gender, and site of care. You are to select 2-5 learning objectives from the following list. The list is in the following format:[Specific Case Guidelines] REQUIREMENTS: *It is important that I am able to distinctly measure each learning objective. However, some of the learning objectives are similar to one another, and, if included for the same communication practice item, it will be difficult to distinguish between what behaviors are exemplifying the different learning objectives. *Similar to the previous requirement, ensure that the learning objectives are not all from the same learning objective CATEGORY. There will be situations when multiple learning objectives from the same category are appropriate, given they can be distinctly measured by different observable behaviors by the physician. After choosing the learning objectives from the following list, review them to ensure that they meet these criteria. Learning objectives: {thelos}

TASK 2 Identify the most appropriate patient affects that the patient might plausibly be exhibiting given the working diagnosis and the communication task. REQUIREMENT: *If the communication task is not about addressing the patient's emotion, only pick possible affects that are more subtle or neutral. The patient should not have a strong affect if the diagnosis and communication task do not indicate it.* Here are the possible patient emotions: {pt_emo}

OUTPUT STRUCTURE The structure of your output should be as follows: DIAGNOSIS: the

exact diagnosis provided CTASK: the communication task as provided AGE: the age range of the patient GENDER: the plausible genders of the patient SOC: the plausible site(s) of care of the patient LOS: the 2-4 identified learning objectives (* new information to be amended to the base case information) AFFECT: the possible affect(s) of the patient, given the other case information (* new information to be amended to the base case information)

Generate that output.

Base Case Details Generation (Step 3): You are an expert clinical communication content designer. You are building content for a communication tool that provides medical professionals the opportunity to practice their communication skills. The content build is occurring in a step-by-step process, and this is just one step in the process. Do not generate any other case content beyond what is requested here. Do not modify, expand, or infer additional conditions, as those will be taken care of in subsequent steps. I will provide you with a patient diagnosis and a communication task. The communication task is the purpose of the physician-patient communication (e.g., explaining the purpose of a vaccination, information about a low-risk procedure or medication, speak with the patient about a common or easily treatable diagnosis).

INPUTS I will provide you with: **DIAGNOSIS** : the patient's diagnosis; **CTASK** : the communication task; **AGE** : a plausible age range for the patient; **GENDER** : the plausible gender(s) for the patient; **SOC** : the plausible site(s) of care where the communication task can plausibly take place; **LOS** : the learning objectives that have been identified for the communication scenario; **AFFECT** : the plausible affect(s) for the communication scenario. Here is that base case information: {step2output} Here are additional descriptions of the selected learning objectives: {focallos}

YOUR TASK Your task is to generate the main 'case bundle'. This consists of 4 components. I will describe each of the components in more detail, and then provide examples of case bundles developed by subject matter experts. The components that you generate should adhere to the descriptions below and follow the same general structure (form, content) as the examples.

CASE BUNDLE COMPONENT DESCRIPTIONS

VIGNETTE: [Specific Case Guidelines]
BACKGROUND: [Specific Case Guidelines]
ENCOUNTER POINT: [Specific Case Guidelines]

PROMPT: [Specific Case Guidelines]
***IMPORTANT DETAILS:** [Specific Case Guidelines]

Length and style targets: [Specific Case Guidelines]

Flow and coherence checks: [Specific Case Guidelines]

*Before finalizing each case, review the output for the following details: [Specific Case Guidelines]

EXAMPLES Here are some example case bundles that have been developed. Use these to help guide the content and structure of the case bundle components: {examples}

OUTPUT STRUCTURE The output should be the 4 case components as follows: **VIGNETTE:** the generated vignette **BACKGROUND:** the generated background **ENCOUNTER_POINT:** the generated encounter point **PROMPT:** the generated prompt

Do not include any additional case details except for the 4 case bundle components. Generate that output.

Patient-Centered Behaviors (Step 4): You are an expert clinical communication content designer. You are building content for a tool that helps medical professionals practice communication skills.

Task: (1) Identify Patient-Centered Behaviors (PCBs) for the provided case. PCBs are discrete, case-specific, observable clinician behaviors that demonstrate how each Learning Objective (LO) would be satisfied. (2) Usually provide one PCB per Learning Objective identified for the case; add multiple ONLY if distinct aspects of the same LO must be addressed.

Constraints: - Only produce PCBs; do not generate any other case content. - Do not modify, expand, or infer additional conditions beyond the inputs. - Keep each PCB to the main idea; avoid parenthetical details, rationales, or sub-examples. - Do not include PCBs for Learning Objectives that are not present in the [CASE_BUNDLE].

EXAMPLES: [Specific Case Guidelines]

Inputs: Core case components [CASE_BUNDLE] containing: **VIGNETTE**, **BACKGROUND**, **ENCOUNTER POINT**, **PROMPT**.

Descriptions of the inputs are as follows: **VIGNETTE**: A brief clinical overview that introduces the primary communication issue and orients the physician to the scenario. **BACKGROUND**: Clinical and contextual information leading up to the conversation so the communication moment is understandable and plausible. **ENCOUNTER POINT**: The moment in the physician-patient interaction that places the clinician at a precise conversational moment that naturally triggers the patient's response. **PROMPT**: The patient's authentic, direct quote that launches the communication challenge and creates opportunity to practice the specified learning objectives (LOS). **LEARNING OBJECTIVES**: The learning objectives that the physician should demonstrate during the case. The format of the learning objectives are: [Specific Case Guidelines] - Example cases with completed PCBs [PCB_EXAMPLES] to illustrate how PCBs map to LOs in context. - ****IMPORTANT**: When reviewing these examples, pay close attention to how the **PATIENT CENTERED BEHAVIORS** are examples of the **LEARNING OBJECTIVES** within the context of the other core case content.

Output format: Return an array that lists the PCBs sequentially and pairs each with its LO name: PCB1: [LO_NAME] - PCB text PCB2: [LO_NAME] - PCB text ...continue until all LOs are covered. ****IMPORTANT****: the [LO_NAME] should be in the **EXACT** same format as provided in the [CASE_BUNDLE].

Do not include any additional case information. Here are case details to guide your task: [CASE_BUNDLE] : {casebundle} [LEARNING OBJECTIVES] : {focallos} [PCB_EXAMPLES] : {pcbexamples}”

Exemplar 1 Generation Prompt (Step 5):

Role: You are an expert clinical communication content designer. Your goal is to improve physician–patient communication for physicians by crafting an exemplar physician response that integrates the provided patient-centered behaviors (PCBs) and demonstrates the specified learning objectives (LOs).

Encounter Information: - Diagnosis: {ptdx} - Communication Purpose: {ctask} - Case Bundle (Vignette, Background, Encounter Point, Prompt): {casebundle} - Learning Objectives: {focallos} - Patient-Centered Behaviors (PCBs): {pcbs}

Definitions: - Vignette: A brief clinical overview

introducing the primary communication issue and orienting the physician to the scenario. - **Background**: Clinical and contextual information leading up to the conversation. - **Encounter Point**: The precise conversational moment that triggers the patient's response. - **Prompt**: The patient's direct quote that launches the communication challenge and opportunity to practice the LOs. - **Learning Objectives**: [Specific Case Guidelines]

Your Task: - Write a single, natural-sounding physician reply that immediately follows the patient's Prompt in the Case Bundle. - The reply must exemplify optimal patient-centered communication for physicians to emulate and incorporate every PCB.

Content Requirements: [Specific Case Guidelines]

Technical Requirements: [Specific Case Guidelines]

Communication Style: [Specific Case Guidelines]

Reference Examples; use these to guide the length, content, and structure of the exemplar response: {exemplarexamples}

Quality Checkpoints (self-check before finalizing): 1. Directly addresses the patient's Prompt and immediate concern. 2. Integrates every PCB seamlessly (no labels, no meta-commentary). 3. Matches tone to the patient's emotional state. 4. Stays within 50–100 words in one paragraph. 5. Sounds authentic—what a skilled physician would actually say.

Output Format: - Provide **ONLY** the exemplar response in this format: [EXEMPLAR]: Your generated physician response here

Generate the exemplar response now.

Exemplar 2 Generation Prompt (Step 6):

Role: You are an expert clinical communication content designer. Your goal is to improve physician–patient communication for physicians by crafting an exemplar physician response that integrates the provided patient-centered behaviors (PCBs) and demonstrates the specified learning objectives (LOs). Some exemplars for this clinical case have already been developed, and an additional exemplar is needed to provide an example of another way that the physician could include the PCBs while demonstrating the LOs for the specific communication task. The additional exemplar should incorporate the same PCBs, but should be

structurally different than previously generated exemplars - the new exemplar should use different wording and sentence construction.

(rest of prompt copied from Step 5)

Patient Ethnicity Selection (Step 7): You are an expert clinical communication content designer. Given the case below, select all patient race codes that could appropriately be represented for this scenario. Do not choose only the most common association; include any race code that is plausible and does not risk reinforcing stereotypes. Only exclude a race code if there is a clear, compelling clinical or contextual reason making it implausible, or if using it in this context would likely reinforce common racial stereotypes.

Inputs: - Diagnosis: {ptdx} - Communication Purpose: {ctask} - Case Bundle (Vignette, Background, Encounter Point, Patient Prompt): {case-bundle}

Key definitions: - Vignette: Brief clinical overview introducing the primary communication issue and orienting the clinician to the scenario. - Background: Clinical and contextual information that makes the communication moment understandable and plausible. - Encounter Point: The precise moment in the interaction that naturally triggers the patient's response. - Patient Prompt: The patient's direct quote that launches the communication challenge and enables practice of the specified learning objectives.

Decision rules: - Clinically plausible means the race code does not contradict the diagnosis or case details; do not use epidemiologic prevalence alone as a reason to exclude. - Avoid reinforcing stereotypes; if a race association is commonly stereotyped in the context of this diagnosis or scenario, exclude that code. - If the case does not provide a compelling exclusion reason, include the race code. - If the case explicitly specifies the patient's race, include only that specified code.

Valid race codes: [Specific Case Guidelines]

Output: - Return only the selected race codes, each wrapped in parentheses, in all caps, separated by a single space. - Do not include any explanations or additional text. - If no compelling exclusions are found, include all race codes.

A.3 Item Quality Dimensions

Items were evaluated with respect to six different dimensions:

- **Medical Accuracy** It was important that the generated item present accurate and plausible medical scenarios for the learners to practice patient-centered behaviors.
- **Learning Objective Alignment** The extent to which the vignette background naturally elicits, depends upon, or provides sufficient context for what the learning objectives are intended to measure.
- **Patient-Centered Behavior Alignment** The degree to which the generated PCBs accurately operationalize the learning objectives within the context of the specific clinical vignette.
- **Exemplar Alignment** The degree to which the exemplar responses accurately and naturally instantiate the PCBs (and, by extension, the LOs) in the vignette context. Here "naturally" means in a manner consistent with how an experienced physician would communicate with a patient (appropriate tone, sequencing, level of detail, etc.).
- **No Stereotypes** The generated item does not include or suggest harmful stereotypes pertaining to medical conditions, demographic characteristics (e.g., race, gender, age), or the association between specific groups and particular illnesses or behaviors.
- **Appropriate Difficulty** The item presents a clinical communication scenario that is suitable for a pre-clerkship medical student, requiring only the level of medical knowledge, reasoning, and communication skills typically expected for learners at this level of training.

A.4 Evaluation Questions for Subject Matter Experts

1. What level of change is needed to make this vignette medically accurate?
 - a. No change (factually correct)
 - b. Minimal (modification of one element (e.g., medical condition incorrectly described/ diagnostic test incorrect)
 - c. Moderate (multiple elements within the vignette need to be revised for medical accuracy)
 - d. Major-Not salvageable (changes would cascade throughout the vignette and exemplars)
2. What level of changes are needed for the learning objectives (LO) to align with the vignette?
 - a. N/A- vignette was not salvageable
 - b. No change (leave as is)
 - c. Minimal (LO not aligned with the case.)
 - d. Moderate (2 LOs not aligned with the case.)
 - e. Major (more than 2 LOs not aligned with the case)
3. Do patient-centered behaviors (PCBs) map to Learning Objectives?
 - a. N/A- vignette was not salvageable
 - b. Yes – no change
 - c. No- PCB requires minor revision(s)
 - d. No- Do not use PCB
4. Do the exemplars align with the PCBs? If not, what level of change is needed for better alignment?
 - a. N/A- vignette was not salvageable
 - b. No change
 - c. Minimal (change to one aspect of the exemplar)
 - d. Moderate (Multiple elements of the exemplar need to be revised)
 - e. Major-Not salvageable (several elements of the exemplars would need to be revised)
5. Does this vignette perpetuate any harmful stereotypes?
 - a. N/A- vignette was not salvageable
 - b. Yes (explain in comments)
6. Is this vignette (background, exemplars, and PCBs) at the appropriate level of difficulty for a learner ready to start clerkships?
 - a. N/A- vignette was not salvageable
 - b. Too easy for student
 - c. Appropriate for a student ready to start clerkships
 - d. Above the level of the student

c. No

A.5 Descriptive Statistics by Generation

Method by All Item Dimensions

Category	Method 1a	Method 1b	Method 2a	Method 2b	Total
Medical Accuracy					
None	146 (73.0%)	134 (67.0%)	114 (57.0%)	105 (52.5%)	499 (62.4%)
Minimal	28 (14.0%)	44 (22.0%)	53 (26.5%)	46 (23.0%)	171 (21.4%)
Moderate	9 (4.5%)	9 (4.5%)	13 (6.5%)	19 (9.5%)	50 (6.2%)
Major	17 (8.5%)	13 (6.5%)	20 (10.0%)	30 (15.0%)	80 (10.0%)
LO Alignment					
None	174 (94.6%)	174 (93.5%)	123 (67.6%)	142 (82.1%)	613 (84.6%)
Minimal	8 (4.3%)	10 (5.4%)	33 (18.1%)	19 (11.0%)	70 (9.7%)
Moderate	1 (0.5%)	1 (0.5%)	24 (13.2%)	9 (5.2%)	35 (4.8%)
Major	1 (0.5%)	1 (0.5%)	2 (1.1%)	3 (1.7%)	7 (1.0%)
PCB Alignment					
None	151 (82.5%)	165 (89.2%)	95 (52.8%)	107 (62.2%)	518 (71.9%)
Minimal	32 (17.5%)	20 (10.8%)	79 (43.9%)	56 (32.6%)	187 (26.0%)
Unusable	0 (0.0%)	0 (0.0%)	6 (3.3%)	9 (5.2%)	15 (2.1%)
Exemplar Alignment					
None	113 (61.7%)	102 (55.1%)	121 (67.2%)	120 (69.8%)	456 (63.3%)
Minimal	54 (29.5%)	63 (34.1%)	45 (25.0%)	36 (20.9%)	198 (27.5%)
Moderate	16 (8.7%)	19 (10.3%)	14 (7.8%)	13 (7.6%)	62 (8.6%)
Major	0 (0.0%)	1 (0.5%)	0 (0.0%)	3 (1.7%)	4 (0.6%)
Appropriate Difficulty					
Too Easy	3 (1.6%)	4 (2.2%)	4 (2.2%)	1 (0.6%)	12 (1.7%)
Appropriate	155 (84.7%)	153 (82.7%)	143 (79.4%)	127 (75.6%)	578 (80.7%)
Too Hard	25 (13.7%)	28 (15.1%)	33 (18.3%)	40 (23.8%)	126 (17.6%)
Stereotype Inclusion					
No	178 (97.3%)	180 (97.3%)	178 (98.9%)	167 (99.4%)	703 (98.2%)
Yes	5 (2.7%)	5 (2.7%)	2 (1.1%)	1 (0.6%)	13 (1.8%)

A.6 Full Rater Alignment Agreement by Item and Method

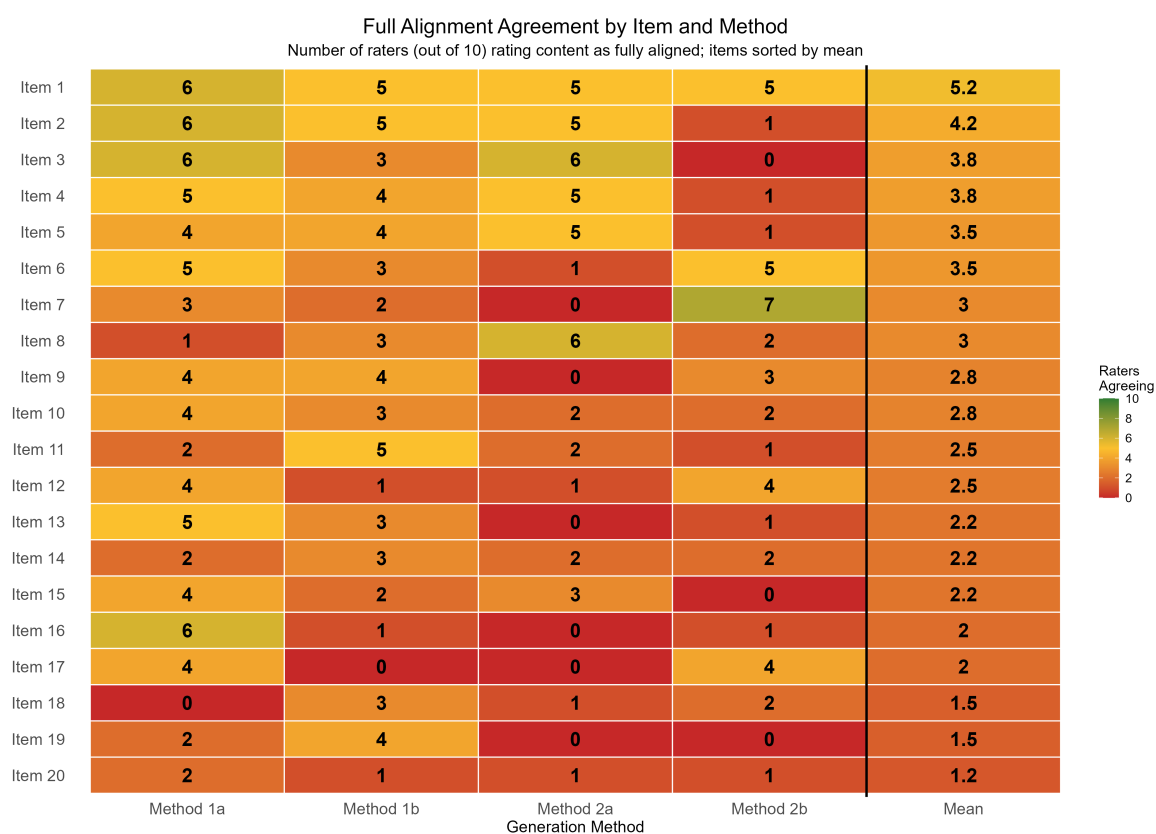


Figure 2: Number of raters indicating that the six item dimensions needed no edits for each generation method and item pair. For example, for Case 17 and Generation Method 1a, only 4 of the 10 raters indicated that no changes were necessary across all six item dimensions.

Towards Self-Referential Analytic Assessment: A Profile-Based Approach to L2 Writing Evaluation with LLMs

Stefano Bannò, Kate Knill, Mark Gales

ALTA Institute, Department of Engineering, University of Cambridge (UK)
{sb2549, kmk1001, mjfg100}@cam.ac.uk

Abstract

Automated essay scoring (AES) research often relies on rank-based correlation metrics to validate analytic assessment. However, such metrics obscure both intrinsic intercorrelations among analytic dimensions that arise from the structure of writing proficiency itself and halo effects, whereby holistic impressions bleed into fine-grained component scores. As a result, high correlations may mask a system’s true diagnostic behaviour. In this study, we propose a novel self-referential assessment evaluation framework that focuses on identifying intra-learner strengths and weaknesses rather than assessing inter-learner rankings. We conduct experiments on the publicly available ICNALE GRA, a uniquely dense second-language writing dataset annotated holistically and analytically by up to 80 trained raters. To obtain reliable reference scores, we apply two-facet Rasch modelling to calibrate rater severity and derive fair average scores across ten analytic aspects and holistic proficiency. We compare the analytic scoring performance of human operational raters and three large language models (LLMs) in a zero-shot setting. Our results show that LLMs tend to outperform single human raters in identifying relative weaknesses (negative feedback) across several proficiency aspects, while human raters remain stronger at identifying relative strengths (positive feedback). Overall, our findings highlight the limitations of rank-based evaluation for analytic assessment and demonstrate the value of intra-learner, profile-based methods for assessing and deploying LLMs in AES.

1 Introduction

Automated essay scoring (AES) has become a cornerstone of second language (L2) proficiency assessment, utilising computational systems to evaluate learner writing in educational contexts. (Klebanov and Madnani, 2022). AES systems have primarily targeted holistic assessment, focusing on

assigning a single score that reflects a learner’s overall writing proficiency. However, driven by the need for more actionable pedagogical insights, the field has recently expanded its focus to analytic scoring, i.e., the evaluation of distinct aspects of proficiency (Li and Ng, 2024). The advent of large language models (LLMs) has revolutionised this landscape. Recent work has successfully exploited LLMs for both holistic (Mizumoto and Eguchi, 2023) and analytic scoring (Bannò et al., 2024). This shift offers the potential for scalable, multi-dimensional assessment that was previously difficult to achieve with traditional supervised models.

Despite these advances, current evaluation practices predominantly rely on rank-based metrics, such as Pearson’s correlation coefficient (PCC), Spearman’s rank coefficient (SRC), or agreement metrics, such as Quadratic Weighted Kappa (QWK) (Yannakoudakis and Cummins, 2015). While these metrics effectively measure a system’s ability to rank learners globally (i.e., inter-learner agreement), they may not be suitable to validate multi-dimensional analytic assessment. In addition to intrinsic intercorrelations among analytic dimensions, a major interfering factor is the halo effect (Thorndike, 1920; Engelhard, 1994), whereby a learner’s general proficiency influences judgments across specific sub-scales. As we show in this study, an LLM can achieve high correlations on analytic aspects simply by acting as a proxy for holistic proficiency, without accurately detecting the nuances that distinguish different analytic proficiency dimensions. Consequently, high rank agreement may not guarantee diagnostic validity. To address this limitation, we propose shifting the analytic evaluation paradigm from *normative* (i.e., ranking learners against each other) to *self-referential* assessment, in which a learner’s analytic skills are evaluated within the same learner profile. Self-referential analytic assessment focuses on intra-learner variability, identifying aspects where

a student performs significantly better or worse than their average proficiency (see Figure 1), thus offering a stricter and more pedagogically relevant test of an LLM’s analytic assessment capabilities.

Ensuring the interpretability of such an evaluation requires a reference signal that is internally consistent, psychometrically calibrated, and representative of expert judgement under the known interdependence of analytic dimensions. Standard datasets often suffer from rater subjectivity and noise, which can obscure the genuine signal. We mitigate this by leveraging the publicly available ICNALE GRA (Ishikawa, 2024) (see Section 3.1), an L2 learner dataset annotated by 80 raters holistically and analytically. We employ Rasch modelling (Linacre, 1989) to rigorously filter raters based on infit statistics, constructing a fair average reference score derived solely from highly consistent human raters.

In this paper, we evaluate three LLMs in a zero-shot setting alongside operational human raters using this self-referential framework. Our contributions are as follows:

1. We demonstrate the limitations of traditional rank-based metrics in analytic assessment, showing that high correlations often mask a lack of diagnostic precision due to both intrinsic intercorrelations and the halo effect.
2. We introduce a novel self-referential analytic assessment evaluation method that disentangles absolute proficiency from relative profile deviations, creating a dual classification task for positive and negative feedback.
3. We report experimental results showing that, within this self-referential framework, LLMs outperform a single operational rater in identifying relative weaknesses (negative feedback), while human raters retain an advantage in identifying relative strengths (positive feedback).

2 Related work

Traditionally, AES systems have mainly focused on overall holistic assessment. This holistic orientation has characterised AES since its inception in the 1960s (Page, 1966) and has been predominant up to today in recent studies investigating L2 assessment with LLMs (Mizumoto and Eguchi, 2023; Yancey et al., 2023).

Research on automated analytic assessment and feedback emerged later, with only a few early exceptions (Page et al., 1997; Shermis et al., 2002). The advent of neural approaches led to a growing body of work targeting specific traits of written production, such as organisation, content, word choice, sentence fluency, and narrativity (Hussein et al., 2020; Mathias and Bhattacharyya, 2020; Ridley et al., 2021; Kumar et al., 2022; Do et al., 2023). As with holistic assessment, LLMs have also been successfully applied to analytic assessment. For instance, Naismith et al. (2023) investigated the use of GPT-4 to predict coherence-related scores. Bannò et al. (2024) explored zero-shot analytic assessment of L2 writing proficiency using GPT-4 across CEFR-aligned proficiency aspects (Council of Europe, 2020). Seßler et al. (2025) examined multiple open- and closed-source LLMs for analytic scoring of German student essays across ten traits, including content-related, syntactic, and formal dimensions. Wang et al. (2025) investigated LLMs’ ability to conduct multi-dimensional analytic writing assessment by assigning scores and generating feedback comments across nine analytic aspects, finding that LLMs can produce generally reliable analytic assessments. In another recent work, Yoo et al. (2025) introduced a new dataset annotated across three analytic traits, i.e., content, language, and organisation, combining LLM-based scoring with synthetic data generation. Using the ICNALE GRA essays, Yamashita (2024) investigated the use of GPT-4 for analytic assessment along complexity, accuracy, and fluency dimensions. Their study primarily adopts a many-facet Rasch measurement framework to analyse rater severity, consistency, and potential biases when treating GPT-4 as an additional rater alongside humans.

Automated analytic scoring has largely inherited the evaluation metrics used for holistic assessment. The studies reviewed above typically rely on rank-based metrics, such as PCC or SRC, as well as agreement measures such as QWK (Yannakoudakis and Cummins, 2015), under a normative assessment framework, whereby learners are ranked against each other for each analytic proficiency aspect. In our work, we propose shifting the evaluation paradigm for analytic assessment from normative evaluation to self-referential interpretation of analytic profiles, in which analytic traits are evaluated within an individual learner’s performance profile. In the psychometric literature, a

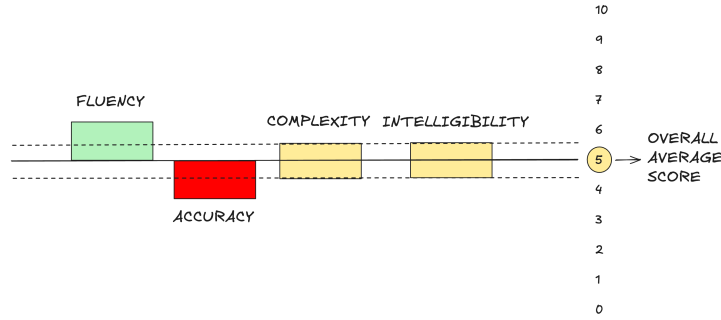


Figure 1: Illustration of proposed self-referential analytic assessment approach.

similar approach has been theorised as *ipsative* assessment (Cattell, 1944; Clemans, 1956). However, in some applied assessment traditions, the same term instead denotes comparison with an individual’s previous performances (Hughes, 2011), rather than within-individual, self-referenced comparison across analytic dimensions. For this reason, we prefer the term *self-referential*.

Self-referential analytic assessment directly supports diagnostic feedback by identifying relative strengths and weaknesses within a learner, rather than across a population. To the best of our knowledge, this is the first study to introduce a self-referential evaluation framework for analytic assessment of L2 proficiency and to implement automated systems for this purpose. The closest conceptual precedent is Berninger and Abbott (2010), who employed within-learner comparisons across language modalities (listening, speaking, reading, and writing), operationalising relative strengths and weaknesses as deviations from a learner’s own mean performance. While their approach focused on modality-level L1 language skills in a psychometric testing context and did not use any automatic systems, the present study applies a self-referential perspective within AES of L2 writing by comparing learners’ performance across fine-grained analytic proficiency aspects.

3 Experimental setup

In this section, we first describe the dataset used in our experiments and outline the procedures for rater selection and score calibration. We then present details on LLM prompting and the method for extracting scores from the models. Next, we discuss the limitations of rank-agreement metrics when evaluating analytic assessment. Finally, we introduce our self-referential analytic assessment system.

3.1 Data

The International Corpus Network of Asian Learners of English (ICNALE) Global Rating Archive (GRA) (Ishikawa, 2020, 2024) is a publicly available L2 learner dataset derived from a subset of ICNALE (Ishikawa, 2011), containing learner essays and speeches, of which we use only the essay section for our experiments. A notable feature of this dataset is its English as a Lingua Franca (ELF) assessment perspective (Seidlhofer, 2005), which evaluates English as a communicative tool among non-native speakers instead of relying on native-speaker norms. Although the dataset is relatively small ($N = 140$), each essay has been annotated by 80 trained raters representing diverse first-language (L1) and occupational backgrounds.¹

In addition to holistic scores, the essays have been annotated across three macro-aspects, which in turn comprise ten analytic rating aspects:

- **Language** (*Intelligibility, Complexity, Accuracy, and Fluency*);
- **Content** (*Comprehensibility, Logicality, Sophistication, and Purposefulness*);
- **Attitude** (*Willingness to communicate and Involvement*).

The analytic scores range from 0 to 10, whereas the holistic scores range from 0 to 100; both scales allow midpoint values.² Human raters were instructed to evaluate the essays holistically first, followed by analytic scoring.

To the best of our knowledge, this makes ICNALE GRA the only publicly available L2 learner

¹Four essays were written by L1 speakers; we did not discard them from the dataset.

²We identified four anomalies in the original scores: for *Intelligibility*, a score of 1.2 (recoded as 1.0) and a score of 6.6 (recoded as 6.5); for *Complexity*, a score of 19 (recoded as 10); and for *Involvement*, a score of 8.6 (recoded as 8.5).

writing dataset that supports a fully crossed rater-essay design with more than two raters, comprising 123,195 ratings.³

3.1.1 Raters selection

In a previous study by the curator of the dataset, focusing specifically on the ICNALE ratings (Ishikawa, 2023), inter-rater reliability is evaluated using Cronbach’s α (Cronbach, 1951). This statistic primarily reflects relative consistency, capturing the extent to which raters produce similar rank orderings of test-takers, while being insensitive to systematic differences in absolute scoring levels. As such, Cronbach’s α is a measure of internal consistency rather than a direct measure of inter-rater reliability (Krippendorff, 2004; Lombard et al., 2006); it reflects the extent to which raters are measuring the same underlying construct.

For this reason, we instead adopt Krippendorff’s α (Krippendorff, 2011), which is more commonly used for inter-rater reliability, as it explicitly quantifies absolute agreement among raters, i.e., the degree to which raters assign identical scores to the same test-taker. Moreover, this metric is well suited to ordinal data, multiple raters, and missing values (see Note 3). Table 1 reports inter-rater reliability in terms of Krippendorff’s α : column *All* shows the estimate computed over all 80 raters, while column *Selected* reports the estimate on 12 raters, after applying the rater selection procedure described below. As can be seen, the low Krip-

Macro-aspect	Aspect	α	
		All	Selected
Language	Intelligibility	0.270	0.531
	Complexity	0.252	0.542
	Accuracy	0.278	0.505
	Fluency	0.256	0.526
Content	Comprehensibility	0.257	0.538
	Logicality	0.237	0.499
	Sophistication	0.225	0.517
	Purposefulness	0.221	0.502
Attitude	Willingness	0.202	0.465
	Involvement	0.172	0.409
	Holistic	0.297	0.579

Table 1: Krippendorff’s α reliability estimates on ICNALE GRA.

pendorff’s α values for *All* indicate substantial disagreement among raters, which may be attributable

³That is, (140 essays $\times$ 80 raters $\times$ 11 proficiency aspects, i.e., ten analytic and one holistic) – 5 missing ratings. Five rating data points are missing from the original scores of the 80 raters; however, there are no missing values in the subset of 12 selected raters (see Section 3.1) used in our experiments.

to differences in severity, inconsistent category use, or divergent interpretations of the rating criteria.

To ensure the reliability of the rating data, we evaluate each of the 80 raters’ consistency using the infit statistic derived from a two-facet Rasch model (Linacre, 1989), i.e., one model for each of the 11 analytic and holistic language aspects, with the two facets being rater severity and learner ability. To do this, we use the Many-Facet Rasch Model estimation function, `tam.mml.mfr`, within R’s TAM (Test Analysis Modules) package.⁴ To accommodate midpoints, analytic scores are fed into the model after multiplying them by 2, whereas holistic scores are binned into 11 categories from 0 to 10, as described in Ishikawa (2024, p. 33).

Infit measures how well a rater’s scores match the model’s expectations, giving more weight to items that are close to a person’s ability level. Mathematically, for rater r and item i , the infit mean-square is computed as:

$$\text{Infit}_r = \frac{\sum_i w_{ri}(X_{ri} - E_{ri})^2}{\sum_i w_{ri}} \quad (1)$$

where X_{ri} is the observed rating, E_{ri} is the Rasch model-predicted rating, and w_{ri} is a weight reflecting the item’s information (i.e., its expected variance). An infit value between 0.5 and 1.5 indicates that a rater’s judgments are consistent with the model, whereas values substantially above 1.5 or below 0.5 indicate over- or under-discrimination, respectively (Linacre, 2002). First, we retained only raters with infit equal to or higher than 0.5 and equal to or less than 1.5, ensuring that all included raters provided reasonably consistent scores across items. Subsequently, we identified the intersection of acceptable raters across all language aspects, resulting in a subset of 12 raters (with 18,480 ratings in total) that satisfied the infit criterion for every aspect considered.

As can be observed in column *Selected* in Table 1, rater selection based on infit statistics leads to a substantial increase in observed reliability. This procedure identifies a subset of raters whose scoring behaviour falls within acceptable Rasch infit bounds and can therefore be used for score calibration and subsequent modelling. From this point onwards, all references to “raters” pertain exclusively to the subset of 12 selected raters.

⁴cran.r-project.org/web/packages/TAM

3.1.2 Score calibration

To obtain a fair average score $E(X_n)$ for a learner n , we fit a two-facet Rasch model for each language aspect, this time only with the 12 selected raters. The net cumulative difficulty barrier, $\Phi_{net,k}$, for achieving a score of k combines the average rater severity ($\bar{\lambda}$) and the global cumulative step difficulties (τ_m):

$$\Phi_{net,k} = \left(\sum_{m=1}^k \tau_m \right) + k \cdot \bar{\lambda} \quad (2)$$

where $\sum_{m=1}^k \tau_m$ is the cumulative step difficulty, defined as 0 for $k = 0$; τ_m is the estimated global step difficulty parameter (the difficulty of moving from score $m - 1$ to m); k : the score category index ($k = 0, 1, 2, \dots, K$); $\bar{\lambda}$ is the mean severity of all selected raters.

The probability of learner n achieving a score k is calculated as the exponent of the difference between the learner’s ability and the net cumulative difficulty, normalised by the sum of probabilities across all possible categories:

$$P(X_n = k) = \frac{\exp(k\theta_n - \Phi_{net,k})}{\sum_{c=0}^K \exp(c\theta_n - \Phi_{net,c})} \quad (3)$$

where θ_n is the estimated ability of learner n and K is the maximum category index.

The final expected score $E(X_n)$ is the sum of the products of each category score s_k and its probability $P(X_n = k)$. Since, for the analytic aspects, the raw scores were multiplied by 2 prior to fitting the model to accommodate midpoints, the final result is divided by 2 for normalisation:

$$E(X_n) = \frac{1}{2} \cdot \sum_{k=0}^K s_k \cdot P(X_n = k) \quad (4)$$

For holistic scores, instead, the final result is

$$E(X_n) = \sum_{k=0}^K s_k \cdot P(X_n = k) \quad (5)$$

We refer to these final scores as *fair average scores*.

Table 2 reports the average Spearman’s rank correlation coefficient (SRC) between the *average rater* (AR) and the reference fair average scores. In other words, we calculated the SRC between each of the 12 raters and the reference fair average scores, and reported the mean and standard deviation. However, in operational language assessment,

essays are typically assigned to raters at random. To simulate this scenario, we randomly selected one rater for each essay. This procedure was repeated five times using different random seeds, and the results were averaged. We refer to this condition as the *operational rater* (OR).

	AR	OR
Intelligibility	0.764 $\pm$.05	0.749 $\pm$.02
Complexity	0.770 $\pm$.04	0.737 $\pm$.02
Accuracy	0.765 $\pm$.05	0.735 $\pm$.02
Fluency	0.761 $\pm$.03	0.733 $\pm$.02
Comprehensibility	0.769 $\pm$.05	0.741 $\pm$.02
Logicality	0.748 $\pm$.04	0.723 $\pm$.02
Sophistication	0.761 $\pm$.03	0.741 $\pm$.04
Purposefulness	0.740 $\pm$.05	0.730 $\pm$.02
Willingness	0.718 $\pm$.03	0.696 $\pm$.05
Involvement	0.692 $\pm$.07	0.644 $\pm$.05
Holistic	0.788 $\pm$.04	0.775 $\pm$.02

Table 2: SRC correlations of average rater (AR) and operational rater (OR) with fair average scores.

Notably, holistic scores exhibit the highest levels of agreement, as shown in Table 1, a pattern that is further supported by the SRC results in Table 2. This observation aligns with previous research indicating that holistic scoring is generally easier and more intuitive for human raters than analytic scoring, hence tending to yield higher inter-rater reliability than fine-grained analytic judgments (Weigle, 2002; Zhang et al., 2015). Furthermore, the *Attitude* aspects (i.e., *Willingness to communicate* and *Involvement*) are inherently more subjective and consequently more difficult for raters to evaluate, as reflected in their lower SRC and Krippendorff’s α values compared with the other aspects.

3.2 LLM prompting and score extraction

We use GPT-4.1 (OpenAI, 2023), Qwen 2.5 72B (4-bit quantised) (Yang et al., 2024), and Llama 3.1 70B (4-bit quantised) (Llama Team, 2024) in a zero-shot setting to evaluate each language aspect in the essays, allowing us to compare both proprietary high-parameter and open-source, medium-size LLMs. The prompt template provided to the LLMs can be found in Appendix A. This contains an analytic rating prompt for each language aspect, originally used by human raters. Because the original analytic prompts (Ishikawa, 2024, pp. 34–36) were designed for both essays and speeches, we adapted them to include only content relevant to

essays. The revised prompts are provided in Appendix B.

To extract proficiency scores from the LLM, we use a weighted average approach (Ma et al., 2025; Bannò et al., 2025). For each analytic aspect, the LLM is prompted independently, producing a probability distribution over discrete proficiency levels via a softmax layer. Let $\mathbf{p} = [p_1, p_2, \dots, p_K]$ denote this probability distribution over K ordinal levels, and let $\mathbf{v} = [v_1, v_2, \dots, v_K]$ represent the numeric values assigned to each level (here, 0 through 9).⁵ The weighted average score for the LLM prediction is then computed as:

$$\text{WAvg} = \sum_{k=1}^K p_k \cdot v_k \quad (6)$$

Intuitively, this procedure produces a continuous estimate of the score by weighting each possible level by the LLM’s predicted probability.

3.3 Limitations of assessing analytic scoring using rank-based metrics

In real-world multi-aspect language assessment, a common characteristic is the high intercorrelation among scores assessing different aspects, as well as the strong correlation between analytic and holistic scores (Lee et al., 2008; Ono et al., 2019). This pattern is also observed in our study. Table 3 presents the SRC between predictions from various systems and the reference fair average scores for three aspects, each representing a macro-aspect: *Complexity* for Language, *Logicality* for Content, and *Willingness* for Attitude. We compare these with the correlations between holistic fair average scores and analytic fair average scores (first row). Additionally, we report the SRC correlations on analytic fair average scores when we prompt our systems for holistic scoring. As can be seen, holistic fair average scores correlate highly with analytic scores. As a result, when we use LLMs prompted to predict holistic scores, we observe strong correlations with analytic scores, in some cases even stronger than when we prompt them for the respective analytic aspect. For example, when prompted for holistic scores, GPT-4.1 shows a correlation with *Complexity* of 0.927 (vs 0.848) or Qwen 2.5 shows a correlation with *Willingness* of 0.754 (vs 0.457). Similarly,

⁵We used a 0–9 scale because including 10 would require two tokens (“1” and “0”), making it difficult to extract a probability for this score.

the operational rater (OR) achieves higher correlations with analytic scores when providing holistic ratings: $0.763_{\pm 0.02}$ for *Complexity*, $0.762_{\pm 0.01}$ for *Logicality*, and $0.750_{\pm 0.02}$ for *Willingness*, compared to $0.737_{\pm 0.02}$, $0.723_{\pm 0.02}$, and $0.696_{\pm 0.05}$, respectively, when performing analytic assessment. In addition to the presence of real intercorrelations among analytic scores, this pattern may be explained by a halo effect, in which the holistic judgments influence ratings across multiple analytic dimensions, effectively acting as a proxy for the learner’s analytic ability.

Consequently, evaluating analytic assessment quality using rank-based metrics such as SRC can be misleading:⁶ these metrics primarily capture global agreement in learner ranking, and may obscure the model’s ability to detect relative strengths and weaknesses within a single learner’s profile. In other words, a system can achieve high correlations with analytic scores simply by reflecting holistic proficiency rather than accurately differentiating performance across individual aspects. It is nevertheless noteworthy that LLMs outperform OR on *Complexity*, *Logicality*, and *Holistic* (with the exception of Qwen 2.5), whereas on the more subjective dimension of *Willingness*, OR still achieves the best – albeit comparably lower – performance, as expected (see end of Section 3.1). The scatterplots for the three selected analytic aspects and holistic proficiency are shown in Figure 3 in Appendix C.

The next section addresses this issue by proposing an alternative approach to analytic assessment.

3.4 Implementing self-referential analytic assessment

We first standardise the reference analytic fair average scores. For each analytic aspect i , we compute

$$z_i = \frac{s_i - m_i}{\sigma_i} \quad (7)$$

where s_i denotes the fair average score, and m_i and σ_i are the mean and standard deviation of that score for i across essays.

For each essay, we then compute the average of the standardised analytic scores across all A aspects:

$$\mu = \frac{1}{A} \sum_{j=1}^A z_j \quad (8)$$

⁶Due to the continuous nature of our scores, we cannot use QWK. However, because this also measures rank agreement, we expect it would yield similar patterns.

		Complexity	Logicity	Willingness	Holistic
Fair Avg. Holistic		0.993	0.982	0.976	1.000
Prompted for analytic	OR	0.737 $\pm$.02	0.723 $\pm$.02	0.696 $\pm$.05	-
	GPT4.1	0.848	0.810	0.340	-
	Qwen2.5	0.753	0.769	0.457	-
	Llama3.1	0.814	0.824	0.554	-
Prompted for holistic	OR	0.763 $\pm$.02	0.762 $\pm$.01	0.750 $\pm$.02	0.775 $\pm$.02
	GPT4.1	0.927	0.921	0.910	0.936
	Qwen2.5	0.738	0.764	0.754	0.771
	Llama3.1	0.758	0.771	0.724	0.781

Table 3: SRC of selected aspects (*Complexity, Logicity, Willingness*) with fair average scores across different systems.

We define the difference between μ and the score for aspect i as

$$\Delta_i = \mu - z_i \quad (9)$$

This difference is used to construct a dual binary classification task corresponding to negative and positive feedback. Specifically, for each aspect i , we define the negative feedback target as $\Delta_i \geq \sigma_{\Delta_i}$ and the positive feedback target as $\Delta_i \leq -\sigma_{\Delta_i}$, where σ_{Δ_i} is the standard deviation of Δ_i across essays.⁷

For model predictions, let $\hat{s}_i$ denote the predicted score for aspect i . Predictions are standardised analogously:

$$\hat{z}_i = \frac{\hat{s}_i - \hat{m}_i}{\hat{\sigma}_i} \quad (10)$$

where $\hat{m}_i$ and $\hat{\sigma}_i$ are computed over the predicted scores for aspect i . The average of the standardised predictions across aspects is then

$$\hat{\mu} = \frac{1}{A} \sum_{j=1}^A \hat{z}_j \quad (11)$$

Finally, we compute the difference between $\hat{\mu}$ and the predicted score for aspect i

$$\hat{\Delta}_i = \hat{\mu} - \hat{z}_i \quad (12)$$

which serves as a continuous scoring signal for the feedback classification task. We report the best $F_{0.5}$ score for each aspect prioritising precision over recall to minimise false positives and avoid misleading learners about their proficiency strengths or weaknesses.

In order to be able to have a fair comparison of the results across the two binary tasks, we

⁷A one-standard-deviation threshold has also been used in the aforementioned work by [Berninger and Abbott \(2010\)](#).

normalise the Precision and $F_{0.5}$ score using the method illustrated in [Raina et al. \(2023\)](#) setting the prevalence to 10% since we are working with different language aspects, each having a different original prevalence rate (see Tables 7, 8, 9, and 10 in Appendix C). To do this, we rescaled the proportion of negative cases so that the effective prevalence matched the target. Let N_+ and N_- denote the number of positive and negative samples, and let pr_{target} denote the desired prevalence. We first solved

$$\frac{N_+}{N_+ + \gamma N_-} = pr_{\text{target}} \quad (13)$$

for the scaling factor γ applied to the number of negatives. Assuming the classifier’s false positive rate per negative instance remains constant, false positives scale proportionally by the same factor γ . The normalised precision P' is then obtained by recomputing precision using this adjusted false-positive count. Normalised recall is unaffected, and the normalised $F_{0.5}$ score is subsequently computed from P' and recall R .

Given this prevalence value, if we calculate the $F_{0.5}$ score for a random classifier, where β is 0.5, P' is 0.10, and R is 1:

$$F_\beta = (1 + \beta^2) \frac{P' \cdot R}{\beta^2 \cdot P' + R} \quad (14)$$

we obtain a score of ~ 12.19 . Any results higher than this value contain information. Table 6 in Appendix C illustrates the percentage of essays receiving feedback across macro-aspects.

Before discussing the results, it is important to clarify the nature of the task. Unlike the inter-learner correlation analyses reported in Table 3, our proposed self-referential framework focuses on identifying aspects that are unusually weak or strong relative to an individual learner’s overall

profile. As a result, systems may show high correlations while still differing substantially in how they distribute strengths and weaknesses across analytic aspects within the same essay.

4 Experimental results

Tables 4 and 5 show the results for GPT-4.1, Qwen 2.5 72B, Llama 3.1 70B, and OR on negative and positive feedback, respectively.

	GPT4.1	Qwen2.5	Llama3.1	OR
Int	44.68	31.96	40.92	40.77 $\pm$ 15.37
Cpl	49.20	37.20	34.00	30.76 $\pm$ 8.01
Acc	24.68	16.75	24.66	26.10 $\pm$ 3.80
Flu	22.52	17.70	20.51	22.37 $\pm$ 5.45
Cpr	37.71	15.71	25.55	34.10 $\pm$ 7.78
Lgc	45.45	36.52	28.18	23.67 $\pm$ 7.67
Sph	20.52	27.71	20.20	23.83 $\pm$ 6.11
Prp	38.20	46.13	51.43	33.25 $\pm$ 9.22
Wil	16.37	12.72	20.11	32.79 $\pm$ 7.61
Inv	37.75	40.51	42.93	45.87 $\pm$ 4.38
Avg.	33.71	28.29	30.85	31.35

Table 4: **Negative** feedback results in terms of best $F_{0.5}$ across systems.

	GPT4.1	Qwen2.5	Llama3.1	OR
Int	21.02	13.80	22.84	35.59 $\pm$ 5.34
Cpl	74.07	65.22	70.00	41.58 $\pm$ 6.13
Acc	29.96	33.61	28.19	34.75 $\pm$ 8.60
Flu	19.44	18.69	22.38	30.23 $\pm$ 4.04
Cpr	23.65	14.11	23.74	29.75 $\pm$ 3.13
Lgc	21.18	37.63	40.52	33.06 $\pm$ 3.64
Sph	37.74	39.47	35.71	29.51 $\pm$ 6.83
Prp	25.68	27.23	37.64	38.23 $\pm$ 5.27
Wil	29.25	29.71	26.89	32.81 $\pm$ 11.11
Inv	37.68	38.98	47.70	40.24 $\pm$ 7.25
Avg.	31.96	31.84	35.56	34.57

Table 5: **Positive** feedback results in terms of best $F_{0.5}$ across systems.

As can be observed, when examining individual analytic aspects under the negative feedback condition (Table 4), LLMs, particularly GPT-4.1, tend to achieve higher $F_{0.5}$ scores than the OR on several *Language* and *Content* dimensions. GPT-4.1 notably outperforms both the other models and OR on *Intelligibility*, *Complexity*, *Fluency*, *Comprehensibility*, and *Logicality*, indicating a stronger alignment with the reference signal in identifying aspects that deviate negatively compared to the overall average score. This suggests that LLMs are especially effective at detecting relative weaknesses in linguistically and structurally grounded dimensions. By contrast, OR attains higher scores on *Accuracy* and on the *Attitude* dimensions (i.e., *Willingness* and *Involvement*), for which all LLMs

show comparatively lower performance. Given that *Attitude* aspects are inherently more subjective and have been shown to exhibit lower inter-rater agreement (see Tables 1 and 2), these differences likely reflect increased noise and variability in the reference labels rather than a systematic limitation of the self-referential assessment formulation or the models themselves. Within the group of LLMs, GPT-4.1 consistently leads across most *Language* and *Content* aspects, while Qwen 2.5 performs best on *Sophistication*, and Llama 3.1 achieves its highest scores on *Purposefulness*. However, overall, GPT-4.1 attains the highest average $F_{0.5}$ score for negative feedback.

For positive feedback (Table 5), a different pattern is observed. While LLMs continue to perform strongly on *Complexity*, *Logicality*, and *Sophistication*, the OR outperforms all models on most *Language* aspects, including *Intelligibility*, *Accuracy*, and *Fluency*, as well as on *Comprehensibility* and *Purposefulness*. Among the LLMs, Llama 3.1 achieves the highest overall average $F_{0.5}$ score under positive feedback, driven primarily by its performance on *Logicality*, *Purposefulness*, and *Involvement*. In contrast, GPT-4.1, while dominant in the negative feedback setting, shows comparatively weaker alignment with the positive feedback targets. Overall, we can observe that LLMs, especially GPT-4.1, exhibit stronger alignment with reference signals for identifying relative underperformance, whereas human raters show higher agreement with the reference in identifying relative strengths. For completeness, in addition to $F_{0.5}$, we also report the results in terms of Precision and Recall in Appendix C in Tables 7, 8, 9, and 10 for OR, GPT-4.1, Qwen 2.5, and Llama 3.1, respectively.

In addition to comparing LLMs with a single randomly selected operational rater per essay, we also report performance for ensembles of multiple raters, ranging in size from 2 to 12. Figure 2 reports the errorbars for feedback on *Logicality*. As can be seen, for this aspect, it takes an ensemble of three raters to outperform the best-performing models, i.e., GPT-4.1 on negative feedback and Llama 3.1 on positive feedback.

For completeness, we report the results for all the aspects and one, two, and three operational raters in Tables 11 and 12 in Appendix C. When averaging performance across aspects, an ensemble of two operational raters is sufficient to outperform the LLMs. However, considering individual aspects, as in Figure 2, larger ensembles are required.

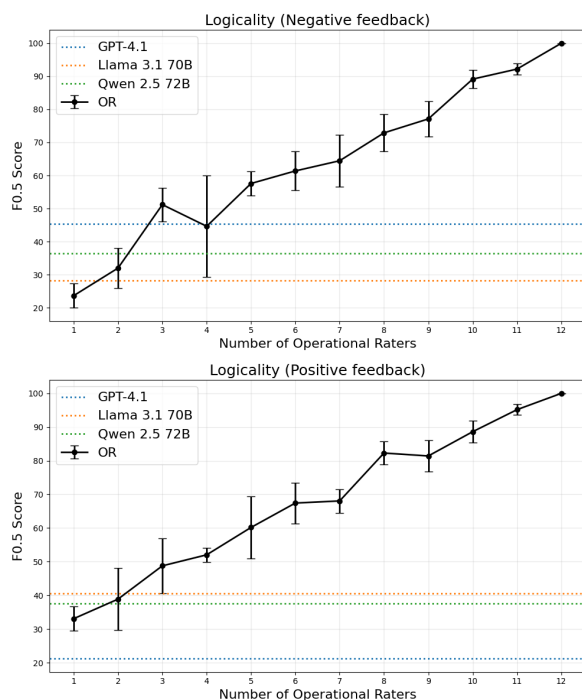


Figure 2: Errorbars on feedback of ensembles with increasing number of operational raters to GPT-4.1, Llama 3.1 70B, and Qwen 2.5 72B for *Logisticality*.

5 Conclusions

This paper addresses the challenges of validating diagnostic analytic assessment in AES, particularly where rank-based metrics may overlook the influence of intercorrelations among analytic dimensions of proficiency as well as the halo effect. Through the introduction of a self-referential framework, we demonstrate how intra-learner evaluation serves as a crucial complement to standard inter-learner assessment.

Our results highlight divergent strengths: LLMs tend to outperform operational human raters in pinpointing relative weaknesses (negative feedback), while humans remain more adept at recognising relative strengths (positive feedback). While the primary focus of this paper is the proposed self-referential assessment framework, we acknowledge that our experiments are limited to zero-shot approaches. Future work will investigate alternative prompting strategies, including few-shot prompting and chain-of-thought reasoning.

Finally, while inter-learner rankings provide essential comparative data, we propose combining them with intra-learner diagnostics. This hybrid approach may offer a pathway to delivering more actionable and informative assessment and feedback to learners, teachers, and testers.

Limitations

First, we acknowledge that relying on a single dataset warrants caution; further experiments are therefore necessary to fully validate the effectiveness of our self-referential assessment framework. While the number of unique essays is relatively small ($N = 140$), the original dataset is exceptionally dense, containing over 120,000 ratings from 80 raters. This unique characteristic of the ICNALE GRA allowed us to implement Rasch modelling in a fully crossed rater-essay design, a rigorous approach not feasible with other L2 learner datasets.

Secondly, as the data focuses on Asian learners within an English as a Lingua Franca (ELF) framework, further research is required to determine if these findings generalise to learners from other L1 backgrounds or for other L2s.

Finally, regarding the self-referential framework, while the use of a one-standard-deviation threshold is a heuristic choice, it ensures a consistent data-driven approach and aligns with established methodology in psychometric literature (Berninger and Abbott, 2010). Moreover, it guarantees that a substantial proportion of learners receive both positive and negative feedback, as shown in Table 6 (Appendix C).

Acknowledgments

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

- Stefano Bannò, Rao Ma, Mengjie Qian, Siyuan Tang, Kate Knill, and Mark Gales. 2025. [Natural Language-based Assessment of L2 Oral Proficiency using LLMs](#). In *10th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 189–193.
- Stefano Bannò, Hari Krishna Vydana, Kate Knill, and Mark Gales. 2024. [Can GPT-4 do L2 analytic assessment?](#) In *Proc. of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 149–164, Mexico City, Mexico. Association for Computational Linguistics.

- Virginia W Berninger and Robert D Abbott. 2010. Listening comprehension, oral expression, reading comprehension, and written expression: Related yet unique language systems in grades 1, 3, 5, and 7. *Journal of educational psychology*, 102(3):635.
- Raymond B Cattell. 1944. Psychological measurement: normative, ipsative, interactive. *Psychological review*, 51(5):292.
- William Vance Clemans. 1956. *An analytical and empirical examination of some properties of ipsative measures (Psychometric Monograph No. 14)*. Psychometric Society, Richmond, VA.
- Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment - Companion volume*. Council of Europe, Strasbourg.
- Lee J Cronbach. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3):297–334.
- 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.
- 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.
- Gwyneth Hughes. 2011. Towards a personal best: A case for introducing ipsative assessment in higher education. *Studies in Higher Education*, 36(3):353–367.
- 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).
- Shin’Ichiro Ishikawa. 2011. A new horizon in learner corpus studies: The aim of the ICNALE project. In *Corpora and language technologies in teaching, learning and research*, pages 3–11. University of Strathclyde Press.
- Shin’Ichiro Ishikawa. 2020. Aim of the ICNALE GRA Project: Global Collaboration to Collect Ratings of Asian Learners’ L2 English Essays and Speeches from an ELF Perspective. *Learner Corpus Studies in Asia and the World*, 5:121–144.
- Shin’Ichiro Ishikawa. 2024. The ICNALE Global Rating Archives: A New Assessment Dataset for Learner Corpus Studies. *Learner Corpus Studies in Asia and the World*, 6:13–38.
- Shin’Ichiro Ishikawa. 2023. Effects of Raters’ L1, Assessment Experience, and Teaching Experience on their Assessment of L2 English Speech: A Study Based on the ICNALE Global Rating Archives. *LEARN Journal: Language Education and Acquisition Research Network*, 16(2):411–428.
- Beata Beigman Klebanov and Nitin Madnani. 2022. [Automated Essay Scoring](#). Springer Nature.
- Klaus Krippendorff. 2004. [Reliability in Content Analysis](#). *Human Communication Research*, 30(3):411–433.
- Klaus Krippendorff. 2011. [Computing Krippendorff’s Alpha-Reliability](#).
- Rahul Kumar, Sandeep Mathias, Sriparna Saha, and Pushpak Bhattacharyya. 2022. [Many Hands Make Light Work: Using Essay Traits to Automatically Score Essays](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1485–1495, Seattle, United States. Association for Computational Linguistics.
- Yong-Won Lee, Claudia Gentile, and Robert Kantor. 2008. Analytic scoring of TOEFL® CBT essays: Scores from humans and e-rater®. *ETS Research Report Series*, 2008(1):i–71.
- Shengjie Li and Vincent Ng. 2024. [Automated essay scoring: Recent successes and future directions](#). In *Proc. of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8114–8122. Survey Track.
- John Michael Linacre. 1989. *Many-faceted Rasch measurement*. Ph.D. thesis, The University of Chicago.
- John Michael Linacre. 2002. What do infit and outfit, mean-square and standardized mean. *Rasch measurement transactions*, 16(2):878.
- Llama Team. 2024. [The Llama 3 herd of models](#).
- Matthew Lombard, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. 2006. [Content Analysis in Mass Communication: Assessment and Reporting of Intercoder Reliability](#). *Human Communication Research*, 28(4):587–604.
- Rao Ma, Mengjie Qian, Siyuan Tang, Stefano Bannò, Kate M. Knill, and Mark J.F. Gales. 2025. [Assessment of L2 Oral Proficiency using Speech Large Language Models](#). In *Interspeech 2025*, pages 5078–5082.
- Sandeep Mathias and Pushpak Bhattacharyya. 2020. [Can Neural Networks Automatically Score Essay Traits?](#) In *Proc. of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91. 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.

- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. [Automated evaluation of written discourse coherence using GPT-4](#). In *Proc. of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.
- Masumi Ono, Hiroyuki Yamanishi, and Yuko Hijikata. 2019. Holistic and analytic assessments of the TOEFL iBT® Integrated Writing Task. *JLTA Journal*, 22:65–88.
- OpenAI. 2023. [GPT-4 Technical Report](#). Preprint, arXiv:2303.08774.
- Ellis B. Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.
- Ellis B Page, John P Poggio, and Timothy Z Keith. 1997. Computer Analysis of Student Essays: Finding Trait Differences in Student Profile. In *Annual Meeting of the American Educational Research Association*, Chicago, IL.
- Vatsal Raina, Nataliia Molchanova, Mara Graziani, Andrey Malinin, Henning Muller, Meritxell Bach Cuadra, and Mark Gales. 2023. Tackling Bias in the Dice Similarity Coefficient: Introducing nDSC for White Matter Lesion Segmentation. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE.
- Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. [Automated cross-prompt scoring of essay traits](#). *Proc. of the AAAI Conference on Artificial Intelligence*, 35(15):13745–13753.
- Barbara Seidlhofer. 2005. [English as a lingua franca](#). *ELT Journal*, 59(4):339–341.
- Kathrin Seßler, Maurice Fürstenberg, Babette Bühler, and Enkelejda Kasneci. 2025. Can AI grade your essays? A comparative analysis of large language models and teacher ratings in multidimensional essay scoring. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 462–472.
- Mark D Shermis, Chantal Mees Koch, Ellis B Page, Timothy Z Keith, and Susanmarie Harrington. 2002. Trait ratings for automated essay grading. *Educational and Psychological Measurement*, 62(1):5–18.
- Edward L Thorndike. 1920. A constant error in psychological ratings. *Journal of applied psychology*, 4(1):25–29.
- Zhengxiang Wang, Veronika Makarova, Zhi Li, Jordan Kodner, and Owen Rambow. 2025. [LLMs can Perform Multi-Dimensional Analytic Writing Assessments: A Case Study of L2 Graduate-Level Academic English Writing](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8637–8663, Vienna, Austria. Association for Computational Linguistics.
- Sara Cushing Weigle. 2002. [Assessing Writing](#). Cambridge Language Assessment. Cambridge University Press, Cambridge.
- Taichi Yamashita. 2024. [An application of many-facet Rasch measurement to evaluate automated essay scoring: A case of ChatGPT-4.0](#). *Research Methods in Applied Linguistics*, 3(3):100133.
- Kevin P. Yancey, Geoffrey Laffair, Anthony Verardi, and Jill Burstein. 2023. [Rating short L2 essays on the CEFR scale with GPT-4](#). In *Proc. of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. [Qwen2.5 Technical Report](#). *arXiv preprint arXiv:2412.15115*.
- Helen Yannakoudakis and Ronan Cummins. 2015. [Evaluating the performance of Automated Text Scoring systems](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–223, Denver, Colorado. Association for Computational Linguistics.
- Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. 2025. [DREsS: Dataset for Rubric-based Essay Scoring on EFL Writing](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13439–13454, Vienna, Austria. Association for Computational Linguistics.
- Bo Zhang, Yunnan Xiao, and Juan Luo. 2015. Rater reliability and score discrepancy under holistic and analytic scoring of second language writing. *Language Testing in Asia*, 5(1):5.

A Appendix A: Prompt

In the context of an examination of English as a Lingua Franca (ELF), a second language (L2) learner of English is asked to write an essay in response to the following prompt:

Do you agree or disagree with the following statement? Use reasons and specific details to support your opinion.

‘It is important for college students to have a part-time job.’

You have to score this essay by only considering the aspect of [ASPECT].

[ANALYTIC RATING PROMPT]⁸

Select a score from 0 (lowest) to 9 (highest). Only output the most suitable score without adding any comment or explanation.

Essay: [ESSAY]

B Appendix B: Analytic rating prompts

Since ICNALE GRA contains essays and speeches, the original analytic rating prompts (Ishikawa, 2024) addressed both modalities. For our experiments with LLMs, we therefore tailored the prompts to retain only the content relevant to essays.

Language

Intelligibility

To which extent can you “decode”, namely, verbally understand what is written? Factors such as spelling and sentence structure may influence it. Please note that intelligibility, which concerns the understandability of the language, should be discriminated from comprehensibility, which concerns the understandability of the content. You may sometimes find an essay that is intelligible but not comprehensible, such as a logically nonsense statement. Meanwhile, you may usually not find an essay that is comprehensible but not intelligible because if the text cannot be decoded, its content cannot be conveyed.

Complexity

To what extent do you think the writer uses morphologically and/or semantically complex words, phrases, expressions, constructions, and grammar? Complexity is seen at many levels of language. For example, “I speculate...” usually sounds more complex than “I think” (Vocabulary). “It is speculated that...” may sound more complex than “I speculate” (Voice, Construction). “If I were a bird” may sound more complex than “If I am a bird” (Subjunctive, Grammar).

Accuracy

To what extent do you think the sample is error-free in terms of vocabulary and grammar? In addition, you should examine the elements such as

punctuation. Please note that you should ignore minor and only-once errors, which may be mistakes rather than errors. Please note that the standard for evaluation should be a proficient non-native ELF speaker, not an English native speaker.

Fluency

To what extent do you think the writer is fluent in the essays? Fluency needs to be evaluated in two ways: (a) fluency and (b) disfluency. If someone writes more, the fluency score should increase, while if s/he uses more disfluency markers, the score may decrease. Disfluency markers include unnecessary connectors (and, but, so because) and semantically empty phrases (such as “I think” most typically), etc. Please note that using these disfluency markers once or twice usually does not cause any problems in communication.

Content

Comprehensibility

To what extent can you understand the content of the essay? Please note that comprehensibility, which concerns the understandability of the content, should be discriminated from intelligibility, which concerns the understandability of the language. If a writer presents a logically reasonable idea, the score should increase.

Logicity

To what extent do you think the idea presented in the essay is logical and reasonable? You need to examine whether the reasons and the conclusions are logically connected.

Sophistication

To what extent do you think the ideas presented in the essay are well-sophisticated, critically thought, unique, original, and innovative?

Purposefulness

To what extent do you think the writer consistently and consciously pays attention to the purpose of the task? The participant was requested to persuade a supervisor to allow them to show their own opinion about part-time jobs for college students in an essay. You have to examine whether the participant fully understands the purpose of the task and consistently sticks to it. Purposefulness is closely related to task completion.

⁸See Appendix B.

Attitude

Willingness to communicate

To what extent do you think the writer is willing to communicate? It is possible that a participant with a limited L2 proficiency shows a high level of willingness to communicate (WTC), and it is also possible that a participant with a high L2 proficiency shows quite a low level of WTC. Factors such as the quantity of writing, the number of ideas s/he presents, and the use of various amplifiers (e.g., “very,” “surely,” “definitely,” “I strongly believe,” etc.) may represent the participant’s WTC.

Involvement

To what extent do you think the participant tries to make the reader involved in his/her discourse rather than writing one-sidedly? The factors such as the use of the second-person pronouns (e.g., “You know,” “as you see,” “as you expect,” etc.) and mentioning the reader are usually related to the degree of involvement.

Holistic

To what extent do you think the sample is close to an ideal ELF essay? Raters have to examine each sample and decide the score (0-100) based on the overall judgment of its quality as a professional ELF output. Please note that “9 points”, for example, should be given to someone who you think is a 100% ideal professional ELF user, not to someone who you think is 100% close to English native speakers. Also, please note that the middle point is 5.

C Appendix C: Additional statistics and results

Table 6 reports the percentage of essays receiving positive and negative feedback across macro-aspects.

Figure 3 shows scatterplots comparing GPT-4.1 predictions with reference fair average scores for the three selected analytic aspects and holistic proficiency. For visualisation purposes, GPT-4.1’s predictions were linearly calibrated using the same essay test data.

Tables 7, 8, 9, and 10 show the complete feedback results in terms of Precision, Recall, and $F_{0.5}$ for OR, GPT-4.1, Qwen 2.5 72B, and Llama 3.1 70B, respectively. Tables 11 and 12 show the feedback results in terms of $F_{0.5}$ for one random operational rater (OR), an ensemble of two random

operational raters (OR2), and three random operational raters (OR3).

Macro-aspect	Negative	Positive	All
Language	45.71	47.14	77.86
Content	48.57	49.29	79.29
Attitude	27.14	31.43	54.29
All	81.43	87.14	92.86

Table 6: Percentage of essays receiving feedback across macro-aspects.

	Negative				Positive			
	P	R	$F_{0.5}$	Prev.	P	R	$F_{0.5}$	Prev.
Int	47.54 $\pm$ 28.59	37.89 $\pm$ 9.05	40.77 $\pm$ 15.37	13.57	43.39 $\pm$ 15.12	29.17 $\pm$ 11.18	35.59 $\pm$ 5.34	17.14
Cpl	34.25 $\pm$ 13.44	34.74 $\pm$ 13.56	30.76 $\pm$ 8.01	13.57	60.66 $\pm$ 23.40	23.64 $\pm$ 7.27	41.58 $\pm$ 6.13	15.71
Acc	30.99 $\pm$ 14.06	31.11 $\pm$ 19.26	26.10 $\pm$ 3.80	19.29	68.73 $\pm$ 25.64	13.33 $\pm$ 4.86	34.75 $\pm$ 8.60	17.14
Flu	20.42 $\pm$ 6.94	55.00 $\pm$ 18.71	22.37 $\pm$ 5.45	20.00	48.48 $\pm$ 26.92	20.77 $\pm$ 10.20	30.23 $\pm$ 4.04	18.57
Cpr	51.23 $\pm$ 26.36	25.45 $\pm$ 12.40	34.10 $\pm$ 7.78	15.71	33.10 $\pm$ 6.91	24.17 $\pm$ 5.53	29.75 $\pm$ 3.13	17.14
Lgc	24.55 $\pm$ 9.37	40.00 $\pm$ 24.58	23.67 $\pm$ 7.67	15.0	62.56 $\pm$ 21.50	17.39 $\pm$ 13.19	33.06 $\pm$ 3.64	16.43
Sph	41.78 $\pm$ 30.56	24.54 $\pm$ 26.44	23.83 $\pm$ 6.11	15.71	33.01 $\pm$ 9.76	34.00 $\pm$ 23.54	29.51 $\pm$ 6.83	14.29
Prp	38.72 $\pm$ 8.42	22.86 $\pm$ 9.71	33.25 $\pm$ 9.22	15.00	64.44 $\pm$ 29.71	22.86 $\pm$ 10.17	38.23 $\pm$ 5.27	15.00
Wil	46.37 $\pm$ 28.31	26.15 $\pm$ 11.77	32.79 $\pm$ 7.61	18.57	33.29 $\pm$ 13.41	42.61 $\pm$ 20.83	32.81 $\pm$ 11.11	16.43
Inv	62.20 $\pm$ 9.49	24.61 $\pm$ 7.54	45.87 $\pm$ 4.38	9.29	63.79 $\pm$ 30.62	28.33 $\pm$ 14.04	40.24 $\pm$ 7.25	17.14

Table 7: Feedback results in terms of best Precision, Recall, and $F_{0.5}$ (OR).

	Negative				Positive			
	P	R	$F_{0.5}$	Prevalence	P	R	$F_{0.5}$	Prevalence
Intelligibility	54.11	26.32	44.68	13.57	18.51	45.83	21.02	17.14
Complexity	73.89	21.05	49.20	13.57	100.00	36.36	74.07	15.71
Accuracy	21.42	62.96	24.68	19.29	31.52	25.00	29.96	17.14
Fluency	24.10	17.86	22.52	20.00	17.80	30.77	19.44	18.57
Comprehensibility	41.70	27.27	37.71	15.71	20.59	58.33	23.65	17.14
Logicity	100.00	14.29	45.45	15.00	18.44	52.17	21.18	16.43
Sophistication	18.50	36.36	20.52	15.71	37.21	40.00	37.74	14.29
Purposefulness	36.40	47.62	38.20	15.00	32.08	14.29	25.68	15.00
Willingness	15.78	19.23	16.37	18.57	32.02	21.74	29.25	16.43
Involvement	44.88	23.08	37.75	9.29	47.23	20.83	37.68	17.14

Table 8: Feedback results in terms of best Precision, Recall, and $F_{0.5}$ (GPT-4.1).

	Negative				Positive			
	P	R	$F_{0.5}$	Prevalence	P	R	$F_{0.5}$	Prevalence
Intelligibility	32.05	31.58	31.96	13.57	11.38	91.67	13.80	17.14
Complexity	36.15	46.11	37.20	13.57	100.00	27.27	65.22	15.71
Accuracy	13.88	96.30	16.75	19.29	34.94	29.17	33.61	17.14
Fluency	15.09	57.14	17.70	20.00	17.02	30.77	18.69	18.57
Comprehensibility	37.34	27.27	34.77	15.71	11.70	79.17	14.11	17.14
Logicity	33.50	57.14	36.52	15.00	53.06	17.39	37.63	16.43
Sophistication	37.34	13.64	27.71	15.71	66.67	15.00	39.47	14.29
Purposefulness	71.58	19.05	46.13	15.00	28.24	23.81	27.23	15.00
Willingness	10.47	92.31	12.72	18.57	36.11	17.39	29.71	16.43
Involvement	68.46	15.38	40.51	9.29	36.94	50.00	38.98	17.14

Table 9: Feedback results in terms of best Precision, Recall, and $F_{0.5}$ (Qwen 2.5 72B).

	Negative				Positive			
	P	R	$F_{0.5}$	Prevalence	P	R	$F_{0.5}$	Prevalence
Intelligibility	67.98	15.79	40.92	13.57	19.96	54.17	22.84	17.14
Complexity	34.67	31.58	34.00	13.57	100.00	31.82	70.00	15.71
Accuracy	23.66	29.63	24.66	19.29	30.92	20.93	28.19	17.14
Fluency	19.16	28.57	20.51	20.00	23.34	19.23	22.38	18.57
Comprehensibility	28.43	18.18	25.55	15.71	20.28	75.00	23.74	17.14
Logicity	26.46	38.10	28.18	15.00	44.17	30.43	40.52	16.43
Sophistication	22.96	13.64	20.20	15.71	100.00	10.00	35.71	14.29
Purposefulness	59.50	33.33	51.43	15.00	44.04	23.81	37.64	15.00
Willingness	21.78	15.38	20.11	18.57	31.14	17.39	26.89	16.43
Involvement	40.85	52.85	42.93	9.29	61.70	25.00	47.70	17.14

Table 10: Feedback results in terms of best Precision, Recall, and $F_{0.5}$ (Llama 3.1 70B).

	OR	OR2	OR3	GPT4.1	Qwen2.5	Llama3.1
Int	40.77 $\pm$ 15.37	34.06 $\pm$ 3.71	45.30 $\pm$ 12.82	44.68	31.96	40.92
Cpl	30.76 $\pm$ 8.01	41.77 $\pm$ 9.13	47.32 $\pm$ 6.78	49.20	37.20	34.00
Acc	26.10 $\pm$ 3.80	33.71 $\pm$ 3.47	44.75 $\pm$ 8.62	24.68	16.75	24.66
Flu	22.37 $\pm$ 5.45	41.12 $\pm$ 18.23	44.13 $\pm$ 6.46	22.52	17.70	20.51
Cpr	34.10 $\pm$ 7.78	38.42 $\pm$ 14.26	41.35 $\pm$ 9.32	37.71	15.71	25.55
Lgc	23.67 $\pm$ 7.67	32.01 $\pm$ 6.07	51.20 $\pm$ 5.14	45.45	36.52	28.18
Sph	23.83 $\pm$ 6.11	23.64 $\pm$ 5.44	40.02 $\pm$ 9.57	20.52	27.71	20.20
Prp	33.25 $\pm$ 9.22	44.81 $\pm$ 8.60	59.79 $\pm$ 9.10	38.20	46.13	51.43
Wil	32.79 $\pm$ 7.61	33.16 $\pm$ 4.57	38.74 $\pm$ 10.10	16.37	12.72	20.11
Inv	45.87 $\pm$ 4.38	49.22 $\pm$ 15.24	52.20 $\pm$ 10.90	37.75	40.51	42.93
Avg.	31.35	37.19	46.48	33.71	28.29	30.85

Table 11: Negative feedback results in terms of best $F_{0.5}$. OR, OR2, OR3 vs GPT4.1, Qwen2.5, Llama3.1.

	OR	OR2	OR3	GPT4.1	Qwen2.5	Llama3.1
Int	35.59 $\pm$ 5.34	35.59 $\pm$ 5.60	39.92 $\pm$ 5.71	21.02	13.80	22.84
Cpl	41.58 $\pm$ 6.13	36.92 $\pm$ 5.66	56.44 $\pm$ 11.69	74.07	65.22	70.00
Acc	34.75 $\pm$ 8.60	45.15 $\pm$ 7.61	50.28 $\pm$ 8.58	29.96	33.61	28.19
Flu	30.23 $\pm$ 4.04	33.46 $\pm$ 7.41	42.79 $\pm$ 5.08	19.44	18.69	22.38
Cpr	29.75 $\pm$ 3.13	30.29 $\pm$ 6.46	45.63 $\pm$ 10.32	23.65	14.11	23.74
Lgc	33.06 $\pm$ 3.64	38.87 $\pm$ 9.27	48.76 $\pm$ 8.13	21.18	37.63	40.52
Sph	29.51 $\pm$ 6.83	28.29 $\pm$ 4.51	36.85 $\pm$ 9.96	37.74	39.47	35.71
Prp	38.23 $\pm$ 5.27	35.08 $\pm$ 5.54	45.71 $\pm$ 13.48	25.68	27.23	37.64
Wil	32.81 $\pm$ 11.11	50.18 $\pm$ 10.59	50.21 $\pm$ 12.03	29.25	29.71	26.89
Inv	40.24 $\pm$ 7.25	51.79 $\pm$ 6.22	61.79 $\pm$ 6.49	37.68	38.98	47.70
Avg.	34.57	38.56	47.84	31.96	31.84	35.56

Table 12: Positive feedback results in terms of best $F_{0.5}$. OR, OR2, OR3 vs GPT4.1, Qwen2.5, Llama3.1.

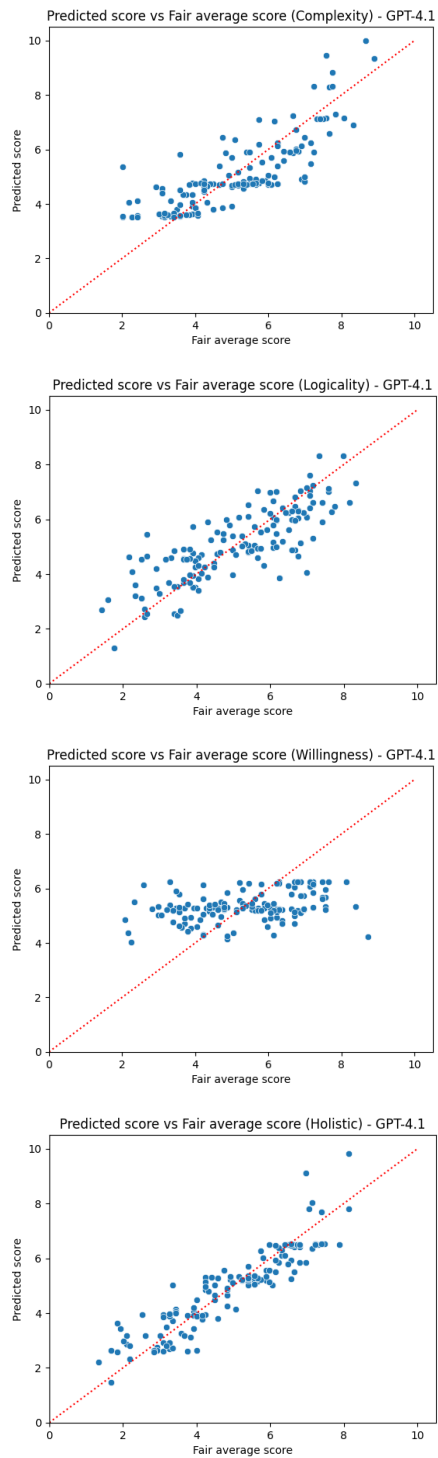


Figure 3: Scatterplots between fair average scores and GPT-4.1 scores for selected aspects.

Using Interaction Log Data to Evaluate and Improve Feedback Accuracy in an Intelligent Language Tutoring System

Mariia Soliar¹, Leona Colling^{1,2}, Stephen Bodnar³, Detmar Meurers^{1,2,3}

¹Leibniz-Institut für Wissensmedien (IWM) Tübingen, Germany
{m.soliar, l.colling, d.meurers}@iwm-tuebingen.de

²Department of Linguistics, University of Tübingen, Germany

³Tübingen Center for Digital Education (TüCeDE), University of Tübingen, Germany
stephen.bodnar@uni-tuebingen.de

Abstract

Intelligent Tutoring Systems (ITS) can record learner interactions in fine-grained detail at scale. This opens the door to data-driven methods for investigating system performance and identifying points for improvement. In this paper, we draw on authentic log data from an English language ITS ($N_{logs} = 5646$, $N_{students} = 368$) to investigate the performance of its feedback algorithm. In step 1 of our analysis, we profiled feedback accuracy by exploring how well the system provided error-specific feedback to malformed student answers in gap-filling grammar exercises using an expert-created set of feedback generation rules. We then identified frequently occurring student errors that triggered incorrect or unspecific feedback and refined the rule set used to detect and respond to these errors with correct specific feedback. In step 2, we validated the rule modifications on an unseen dataset. Comparing the performance of the initial and updated rule sets, we find significant improvement that generalizes to unseen data. Our study thus illustrates how an empirical evaluation of authentic data can complement feedback creators' expertise by informing rule refinement decisions that yield significant and generalizable performance improvements to feedback in ITS systems.

1 Introduction

Among other trending approaches to computer-assisted language learning (CALL, [Huang et al., 2023](#); [Zerkouk et al., 2025](#)), Intelligent Tutoring Systems (ITS) have demonstrated a notable capacity to improve learning outcomes ([Heift, 2010](#); [Meurers et al., 2019](#)), enhance learner engagement, motivation, and foster positive social outcomes ([Ukenova and Bekmanova, 2023](#)). Both educational practice and Skill Acquisition Theory ([DeKeyser, 2007](#)) motivate the need for deliberate practice with feedback, which can be facilitated

by performing error detection and providing automatic corrective feedback ([Amaral et al., 2011](#); [Ruiz et al., 2023](#)).

Feedback plays a crucial role in language acquisition, and therefore its correctness and relevance is of high importance ([Loewen, 2013](#)). When it comes to feedback in ITSs, [Cavalcanti et al. \(2021\)](#), in their review of automatic feedback generation systems in online learning environments, concluded that automatic feedback can be as effective, or sometimes even more effective, than manual instructor feedback.

Another beneficial property of ITSs is the ability to record large amounts of user-system interactions, enabling educational data mining to draw conclusions about both learner performance and system behavior ([Romero and Ventura, 2013, 2020](#)) and, therefore, to evaluate the performance of feedback generation ([Koedinger et al., 2013](#)).

In recent years, the focus of using AI in education has shifted significantly towards Large Language Models (LLMs) and the integration of generative AI ([Dong et al., 2024](#); [Degraeuwe and Goethals, 2024](#); [Baidoo-Anu and Ansah, 2023](#); [Dai et al., 2023](#)). Research shows that LLMs can generate useful formative feedback and the accuracy of the latter is promising, but false positives are still a concern, especially in educational settings, as they might undermine learners' confidence and lead to misconceptions ([Jaganov et al., 2025](#)). Moreover, for closed exercise types and semi-open exercises (e.g., gap filling with grammatical forms) where the set of correct answers is known, a rule-based or hybrid approach to generating feedback can follow the "don't guess if you know" advice for achieving transparency and robustness.

Therefore, as [Stamper et al. \(2024\)](#) suggest, one should not disregard previous knowledge and years of experience using expert-generated and traditional data-driven approaches that allow high

levels of control over feedback provision, a system feature that is particularly crucial for beginner learners who tend to rely heavily on system feedback (Fu et al., 2022; Yang and Li, 2024).

In this paper, we analyze feedback generation performance of the FeedBook (Rudzewitz et al., 2017), a language ITS that primarily focuses on semi-open gap-filling grammar exercises (e.g., "[...] he now? He is at home now". Target answer: "Where is") and provides dynamic scaffolding feedback. We aim to showcase the value of authentic interaction logs for evaluating and refining a rule-based feedback system, such as the one in the FeedBook by addressing the following research questions (RQs):

1. How can systematic analysis of authentic logs be applied to evaluate and improve rule-based feedback generation?
2. How well does the approach generalize to unseen data?

The results of our experiment show that log analysis can be used effectively to identify rules to refine or add, and that even a small number of changes can yield significant performance improvements. We also employ statistical analysis to demonstrate the generalizability of the approach to unseen data.

2 Related Work

ITSs play an increasingly significant role in modern education, and therefore, the importance of evaluating these systems to ensure their technical reliability, usability, and overall effectiveness increases proportionally (Mousavinasab et al., 2021; Marouf et al., 2024; Latif et al., 2026).

Siemer and Angelides (1998) were among the first to specify an evaluation framework for ITSs and suggested a two-part evaluation: external and internal. External evaluation is concerned with the system’s influence on the student, e.g., on learning achievement and learning affect. As for internal evaluation, the goal is to analyze the system from within, to evaluate its functionality and efficiency.

Most ITS evaluations (e.g., Bór, 2023; Wu et al., 2022; Erümit et al., 2019; Karaci et al., 2018) focus on external system effectiveness by analyzing usability, learner performance, or motivation rather than evaluating the intelligent algorithms employed. When it comes to feedback evaluation and enhancement in ITSs, Banihashem et al.

(2022) confirm that learning achievement and pedagogical gains are the primary goals of such studies. For example, Van der Kleij et al. (2015) analyzed the effect of more elaborate feedback on learning outcomes compared to simpler feedback and the provision of correct answers. Lavolette et al. (2015) analyzed the accuracy of error detection and the respective feedback but the study focused on the effect of immediate feedback versus delayed feedback and not on fine-tuning or enhancement of the feedback itself. Inn-Chull (2019) and Fazilatfar et al. (2024) explored the effect of computer-assisted feedback on learner uptake and analyzed learner data to tailor feedback strategies (e.g., example-based or metalinguistic) based on their pedagogical effectiveness. Horbach et al. (2022) also examined the effect of automated informative tutorial feedback on learner writing performance.

Today, research often focuses on LLM-generated feedback, mainly featuring short answer and free writing assessment (Stahl et al., 2024; Jaganov et al., 2025; Emirtekin, 2025). The approaches to providing grammar feedback in these studies are partially inspired by Grammatical Error Correction (GEC) systems, and a significant amount of recent work has evaluated them (e.g., Volodina et al., 2023; Kobayashi et al., 2024). However, in their comprehensive survey Bryant et al. (2023) emphasize that most GEC systems focus on correcting errors without explaining them (e.g., Omelianchuk et al., 2024), which limits their educational value. Although some resources for generating explanatory feedback are emerging (e.g., Fei et al., 2023; Ye et al., 2025), more work is needed to develop robust, explainable GEC and to evaluate the generated feedback (Nagata, 2019; Nagata et al., 2020, 2021).

When it comes to data, Leacock et al. (2014) highlighted the importance of authentic learner texts for training GEC and feedback tools, enhancing their effectiveness in CALL and automatic writing evaluation contexts. This benefit has recently been emphasized, for example, by Song et al. (2024) in their work on Grammar Error Explanation. They propose a two-step pipeline that uses fine-tuned and prompted LLMs trained on real-world learner corpora in order to first edit erroneous input and subsequently prompt GPT-4 for explanations of the edits.

Greer and Mark (2016) also highlighted the benefits of analyzing the fine-grained data col-

lected by an ITS for its evaluation. For instance, Heift (2013) used learner interaction logs to improve the *Error Priority Queue*, i.e., to determine the relevance of feedback messages to the learning goal whenever several errors were present in student answer.

Interestingly, our research on semi-open gap-filling grammar exercises revealed that there are very few works on feedback generation for such exercises or evaluation thereof apart from the work by Rudzewitz et al. (2018). Although free production is the ultimate goal of language learning, constrained constructed responses - such as gap-fills - are effective in building explicit language knowledge and accuracy (Ellis, 2006). This learning effect, as mentioned above, can be enhanced with scaffolding feedback. However, most research involving gap-filling is concerned with the generation of the questions and the gaps themselves (e.g., Kiros Bitew et al., 2023; Hill and Simha, 2016; Knoop and Wilske, 2013), or distractors for those gaps (e.g., Yoshimi et al., 2023; Panda et al., 2022; Yeung et al., 2019), rather than feedback. These studies are also rarely associated with an ITS.

To address this gap, we adopt the data-mining approach and use authentic logs of learner interactions to perform an internal evaluation of the feedback mechanism in the FeedBook and demonstrate the benefits of the approach for gap-filling exercises.

3 Feedback Mechanism in FeedBook

The FeedBook is a web-based ITS designed for 7th grade English as a second language (ESL, L2) learners in German secondary schools. It is publicly available for use in ESL classrooms and is under ongoing development within research projects.¹ Its main function is to provide learners with language exercises and to give immediate, individualized scaffolding feedback (Rudzewitz et al., 2017). Feedback becomes available in two stages: offline *feedback generation* and real-time *feedback provision* (see Rudzewitz et al. (2018) for a detailed description of the algorithm).

FeedBook’s **feedback generation** mechanism utilizes various Natural Language Processing (NLP) tools, such as Apache Unstructured Information Management application (UIMA) (Ferrucci et al., 2009), ClearNLP (Choi and Palmer,

2012), Morpha (Minnen et al., 2001), and Sfst (Schmid, 2005) to analyze and annotate exercises, including the encoded target answer (TA; i.e., the expected correct answer), in order to predict possible learner errors and provide appropriate and helpful feedback options. TAs can be full but single sentences, phrases or single words. If a TA comprises a partial sentence, the remaining parts (i.e., to the left and right of the gap) are provided as *context* to the NLP pipeline. To account for the phenomenon of multiple admissibility (Katinskaia et al., 2019), some exercises have one or several alternative TAs.

The system utilizes a multi-layer rule-based approach (see Appendix A) that takes TAs and transforms them into ill-formed or well-formed but otherwise incorrect anticipated learner answers (*hypotheses*). The transformations are made through a variety of second language acquisition (SLA) theory-based, pedagogically driven rules, e.g.,

overregulation of irregular verbs in simple past: *did* > *doed*, *went* > *goed*;
incorrect tense or person agreement of auxiliary verbs: *do* > *did*, *was* > *were*.

Each hypothesis carries the transformed surface string information, the respective error diagnosis, and the context and matches it with the most specific feedback message for later display to learners.

In terms of **feedback provision**, once a learner enters an erroneous answer, the system compares it with the generated hypotheses, identifies the closest match and displays its associated feedback message as a popup. The messages are teacher-crafted metalinguistic error explanations tailored to our target population, e.g., “*You’ve forgotten an auxiliary verb (‘Hilfsverb’, e.g. ‘do’/‘did’).*”. The system provides one feedback message at a time, adjusting it when a learner attempts to correct their answer. The messages cover a wide range of grammar errors, spelling and capitalization errors. If the system cannot diagnose the error, it falls back to a *default feedback* message: “*This is not what I’m expecting. Please, try again.*”.

4 Evaluation Approach

In this paper, we propose an approach to systematically use authentic data to enhance a rule-based feedback system for semi-open gap-filling grammar exercises and evaluate the generalizability of

¹The system and information about the related projects are available at <https://feedbook.website/>.

improvements. The approach uses two datasets: a training dataset to identify refinement points in a data-driven way, and an unseen testing dataset to assess generalizability.

As shown in Figure 1, we suggest first establishing a baseline by processing authentic data through a pipeline that mimics the feedback system and then annotating the output to evaluate its correctness and granularity. The analysis focuses on three aspects: 1) identifying and correcting faulty rules to improve precision ², 2) extending existing rules to increase recall, and 3) introducing previously missing rules to further improve recall.

After implementing the identified changes and adjusting the system accordingly, the data is processed again to establish improvements in precision and recall. Once a significant improvement has been confirmed, we suggest testing the updated system on unseen data to avoid overfitting.

We demonstrate the approach using the FeedBook, its feedback mechanism, and data from its use in real classrooms.

4.1 Data

The FeedBook has been used in German schools for four large-scale field trials, each focusing on a different aspect of the system. For our analysis, we drew on user data from two consecutive trials (Parrisius et al., 2022; Berens et al., 2024). ³

For a concise analysis, we focused on a single learning target, ‘Questions’, selected due to the complexity of interrogative syntactic structures, their known difficulty for L2 learners, and their presence at all stages of learning (Mackey, 1999). We included only data units from semi-open gap-filling exercises, where typed student answers (SAs) provided valuable insight into actual learner needs by revealing weaknesses in students’ productive grammatical skills (see Appendix E for examples of the exercises). Each log data unit contains SA, TA, and *context*:

SA (userInput): *is*
TA (targetAnswer): *Was*
context: *he late? Yes, he was late.*

We removed units with correct SAs, fully duplicate units, and units where capitalization was the

only issue with the SA as such cases were considered correct during the second data collection.

The two data collections served as the *training* and the unseen *testing* dataset, respectively. For a more precise analysis, we divided each dataset into three subsets, based on the TA length: Word (e.g., *Where, do, is*), Phrase (e.g., *did you do*), and Sentence (e.g., *What does Jenny want to do?*).

In the second data collection, the number of users increased, leading to a much larger number of data units. To simulate one iteration of 10-fold validation per subset, we extracted a stratified random sample for the testing data (see Appendix B). Table 1 summarizes the final dataset sizes. ⁴

	Training			Testing		
	Units	TAs	Ex.	Units	TAs	Ex.
Wrd	720	48	15	80	26	8
Phr	2083	60	14	236	43	14
Sent	2273	48	8	254	43	8
Total	5076	156	37	570	112	30

Table 1: The sizes of training and testing datasets. Wrd: word, Phr: phrase, Sent: sentence, TAs: target answers, Ex.: exercises.

The training dataset contains 5076 data units from 267 students and 37 exercises. In the Sentence subset, seven TAs have alternative correct answers (e.g., *Where are you from?* for *Where do you come from?*). They are not counted as separate TAs but accounted for in the feedback algorithm and our evaluation environment (see Section 4.2). The testing dataset contains 570 data units from 101 students and 30 exercises, with nine and 17 alternative TAs in the Phrase and the Sentence subsets, respectively.

The initial hypothesis generation rule set for questions was inspired by established SLA theories on developmental sequences that postulate that learners acquire grammatical structures in a constrained and ordered manner (Petersen, 2010; Pienemann et al., 1988). These theories provided the conceptual foundation for identifying question-related linguistic phenomena that are both developmentally important and frequently problematic for L2 learners. Building on this foundation, together with an expert teacher, we structurally selected, extended, and prioritized the

²We define precision as provision of incorrect vs. correct specific feedback and recall as specific vs. default feedback.

³Ethical and ministerial approval to collect data for research purposes was granted.

⁴Data is available for download at https://osf.io/3hrud/overview?view_only=ede0c00a7e36489884bd8cf788bd16eb.

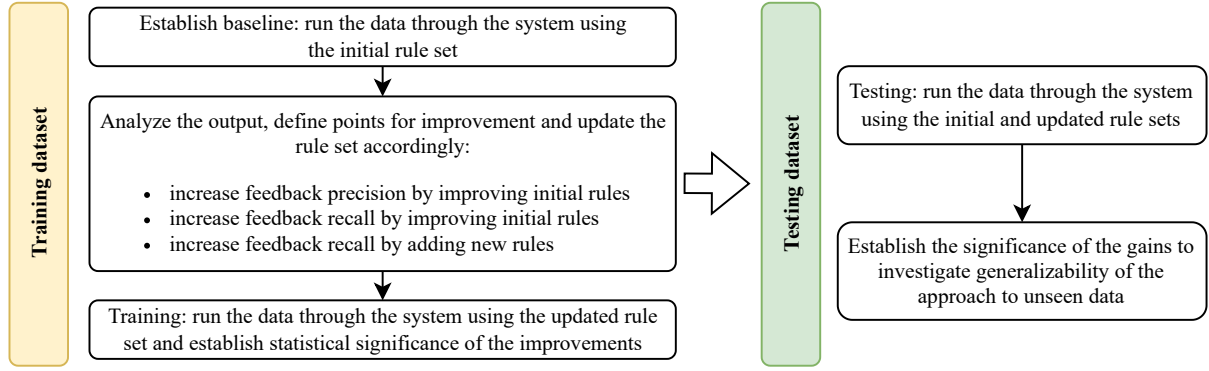


Figure 1: Proposed approach for employing authentic logs to evaluate and improve feedback in an ITS system.

rules. As a result, the initial rule set for questions contained eight rules implementing transformations such as missing or incorrect auxiliary verbs or question words, missing question marks, incorrect tense or person agreement of the auxiliary verb “do”, incorrect form of the main verb when an auxiliary is present (see Appendix C). Some rules rely on syntactic dependency, others on the token position in a sentence.

4.2 Methods

As a first step, we created an environment that simulates FeedBook’s feedback generation and provision including the SA pre-processing steps (e.g., string normalization). Unlike the actual system, our simulation takes one TA-SA pair at a time as input, generates hypotheses for the given TA, and then matches the SA with a feedback message. If the TA has alternatives, both the system and our simulation generate hypotheses for all given TA variations.

We ran four simulations to obtain the following performance data for analysis (S - seen, training data, U - unseen, testing data):

- **baseline_S** (training data with initial rules)
- **update_S** (training data with updated rules)
- **baseline_U** (testing data with initial rules)
- **update_U** (testing data with updated rules)

The output included SA, TA, and a respective fine-grained question-specific (QS) feedback label (e.g., “missing question word”). For this initial analysis, we combined all QS feedback into one score. Based on the baseline_S results, we identified rules to refine or add and implemented the changes. Next, we performed a comparative analysis of baseline_S and update_S expecting

an increased proportion of QS feedback and reduced default feedback. Then we compared the baseline_U with the update_U, expecting the same trend as evidence for generalizability to unseen data.

Two expert annotators manually labeled the fine-grained feedback⁵ to evaluate its correctness and report on precision and recall based on the following annotation schema:

- “optimal” – correct error diagnosis (true positives, see Figure 2 a)); or default feedback provided for nonsensical input (true negatives, see Figure 2 b));
- “incorrect” – incorrect specific diagnosis of an error (false positives, see Figure 2 c));
- “unspecific” – default feedback where a specific one could (with a new rule) or should (with an existing rule) have been given (false negatives, see Figure 2 d)).

It is important to point out that “unspecific”, i.e. default feedback should not be considered incorrect but only a less optimal variant of feedback that the system can provide to the students. The diagnostic process for the feedback provision is optimal if the system correctly identifies the specific error contained in the SA. If the system only detects the presence of an error, the process is still considered successful, as immediate correctness feedback is one of the mechanisms by which ITSs achieve learning gains comparable to human tutors (VanLehn, 2011).

To assess the significance of the impact that rule improvements had on system’s performance,

⁵In case of conflict, the feedback was labeled by a third independent annotator and the majority vote was used for the final scores.

	Data unit	Annotation
a)	SA: Who are you doing? TA: What are you doing? FB: INCORRECT_QW	final_label: optimal
b)	SA: rgzrg TA: What are you doing? FB: DEFAULT	final_label: optimal
c)	SA: What are you always watching? TA: What do you always watch? FB: MISSING_AUX	final_label: incorrect
d)	SA: Do you afraid of dogs? TA: Are you afraid of dogs? FB: DEFAULT	final_label: unspecific

Figure 2: Examples of data units and their respective annotations. SA: student answer, TA: target answer, FB: feedback, QW: question word, AUX: auxiliary verb.

we use McNemar’s test with continuity correction (Agresti, 2012).

5 Baseline Evaluation and Rule Set Enhancement

Table 2 shows the raw numbers of the feedback from the baseline_S evaluation as well as the annotated correctness for all three subsets. The inter-annotator agreement, measured using Cohen’s Kappa, is $\kappa = 0.97$, indicates almost perfect agreement (Cohen, 1960; see Appendix D). The corresponding precision and recall scores are summarized in Table 3.

The baseline_S results revealed that the overall success rate of the feedback provision is very high - 94.37%. However, whilst only a small proportion of errors were diagnosed incorrectly (5.63%), a significant proportion did not receive optimal feedback, i.e. a specific diagnosis (59.83%), indicating a substantial need to improve recall. To address this, we closely analyzed both SAs and TAs in each subset. Based on the distribution of linguistic constructs in TAs (see Table 4), we investigated SAs containing errors related to the targeted constructs and observed the following: in all subsets, errors such as incorrect question word, missing or incorrect auxiliary, and incorrect agreement of auxiliary “do” are the most common and are rarely diagnosed correctly (only 10.92% to 27.12% per error type). We summarize the numbers in Table 5.

Further investigation showed that errors such as incorrect agreement of the main verb “be” or its replacement by auxiliary verbs, or missing main verb were not diagnosed due to the lack of respective rules. Moreover, the Phrase and Sentence subsets contained 54 SAs where only word order was incorrect (e.g., *you do have* instead of *do you have*). It was particularly insightful to observe that such malformations as switching the tense forms of auxiliary and main verbs in simple past questions (e.g., *Do you saw?* instead of *Did you see?*) or the conjugation of “be” with the help of “do” (e.g., *does be* instead of *is*) were more common than our experts anticipated. In addition, students often omitted or added tokens to their answers.

The annotations of baseline_S also showed that most cases labeled as “incorrect” received spelling feedback where it should have been default feedback for nonsense SAs (66.81%) or QS feedback (6.64%), i.e., incorrect question word or incorrect auxiliary verb. Although we addressed the latter issue by enhancing the rule set, improving the spell-checking algorithm (not a part of the hypothesis generation) was not in focus for this paper.

Based on these observations, we refined four initial rules (i.e., missing and incorrect auxiliary verb, incorrect agreement thereof, incorrect question word), and added four new rules introducing transformations such as missing main verb, incorrect tense and person agreement of the verb “be”, verb “be” being replaced by verbs “do” or “have”, switching the tenses of auxiliary “do” and main verb in past simple tense. An additional check to account for word-order permutations was implemented.

It is worth noting that in the Phrase and Sentence subsets, 3 and 18 SAs, respectively, were grammatically and semantically correct but did not match any of the provided TAs. The system diagnosed these cases as spelling or default, and thus, they contributed to the counts for “incorrect” and “unspecific” categories. This observation suggests the refinements should include not only the feedback generation rules but also the alternative correct answers or fuzzy-matching. Addressing this was outside the scope of this paper, but is a promising direction for future work and a perfect phenomenon to explore the benefits of LLM-based or hybrid approaches.

Initial rules									
Feedback	QS		Spelling		DF		Total		
	✓	✗	✓	✗	✓	◆	✓	✗	✓+◆
Word	66	5	132	85	6	426	204	90	630
Phrase	202	42	571	61	33	1174	806	103	1980
Sentence	126	17	587	76	30	1437	743	93	2180
Total	394	64	1290	222	69	3037	1753	286	4790
						(59.83%)	(34.54%)	(5.63%)	(94.37%)
Updated rules									
Word	179	0	132	85	6	318	317	85	635
Phrase	448	1	571	61	33	969	1052	62	2021
Sentence	266	23	587	75	30	1292	883	98	2175
Total	893	24	1290	221	69	2579	2252	245	4831
						(50.81%)	(44.37%)	(4.83%)	(95.17%)

Table 2: Results for initial and updated rules on **training data**: Raw numbers of provided feedback annotated as optimal (✓), incorrect (✗) and unspecific (◆) per subset (word, phrase, and sentence). QS: question-specific, DF: default feedback.

	Precision	Recall
baseline _S	0.86	0.37
update _S	0.90	0.47
baseline _U	0.81	0.35
update _U	0.92	0.55

Table 3: Precision and recall scores.

Ling.constr. in TA	Word	Phr	Sent
QW only	189	-	-
QW + be	25*	161	190
QW + do/does/did	-	493	1368
do/does/did	139	1323	-
be	296	-	-
Y/N + do/does/did	-	-	607
Y/N + be	-	-	108
modals	-	106	-
main verb	71	-	-
Total:	720	2083	2273

Table 4: Numbers of data units, i.e. erroneous SAs, clustered by linguistic constructs focused in the TAs. Top-3 constructs per input type are highlighted in bold. QW: question word, QM: question mark. *25 data units in Word subset were two-word phrases.

Updated Rule Implementation

We employed the syntactic dependency approach in all new rules and in four initial rules that received an update (see Appendix A). In addition, we extended the *Incorrect Auxiliary Rule* by configuring it to generate hypotheses with incorrect

Error	Word	Phr	Sent
Incorr. QW	43 / 115	39 / 161	14 / 78
Missing QW	1 / 3	13 / 60	5 / 23
Incorr. agr. “do”	16 / 42	67 / 333	31 / 175
Incorr. auxiliary	2 / 46	25 / 161	4 / 77
Missing auxiliary	0 / 1	43 / 267	8 / 159

Table 5: Number of selected student errors in baseline_S; diagnosed correctly/total errors made. Top-3 errors per input type are highlighted in bold. QW: question word, agr.: agreement.

grammatical number (the initial rule only transformed the lemma and the tense) and for the *Question Word Replacement Rule* we extended the list of question words based on log data observations.

5.1 Results Update_S

Table 2 shows the raw numbers of the feedback provided in the update_S as well as the annotated correctness for all three subsets. The inter-annotator agreement is $\kappa = 0.98$ (almost perfect agreement). The system with updated rules provided less default feedback (-9.02 percentage points (p.p.)) and more QS feedback (+9.83 p.p.) compared to the baseline_S, suggesting an overall positive effect of the refinements.

Interestingly, the number of “incorrect” cases decreased for the Word and Phrase subsets but increased by six for the Sentence subset. 12 “incorrect” cases in Sentence and one in Phrase received a new “missing main verb” diagnosis in

contrast to the default (or spelling in one case) in the baseline_S. Ten “incorrect” cases in Sentence were diagnosed as “incorrect word order”. These results suggest that the implemented changes are largely beneficial, but might still introduce errors.

Initial rule set	Updated rule set		Total
	✓	✗ + ◆	
✓	1752	23	1775
✗ + ◆	500	2801	3301
Total	2252	2824	5076

Table 6: Contingency table for training data.

Overall, we observed an increase in precision and an even greater increase in recall (see Table 3). The results of McNemar’s test (see Table 6) showed that the improvement of the feedback performance with the updated rule set is statistically significant: McNemar’s $\chi^2 = 433.22$, $df = 1$, p -value $< .0001$, Cohen’s $g = 0.46$ (large effect size).

6 Results of Generalizability Evaluation

Turning now to the testing dataset (U), table 7 shows the raw numbers of the feedback provided in the baseline_U and in the update_U as well as the annotated correctness for all three subsets. The inter-annotator agreement for the two runs are $\kappa = 0.96$ and $\kappa = 0.97$, respectively (almost perfect agreement). The updated rules consistently produced less default feedback (-16.85 p.p.) and more QS feedback (+14.46 p.p.) compared to baseline_U. Although some QS cases became incorrect with the update, the overall “incorrect” feedback decreased (-3.15 p.p.). The corresponding precision and recall scores are summarized in Table 3. These results suggest a positive trend and confirm the observations on the training data.

The results of McNemar’s test (see Table 8) show statistical significance of the improvements on unseen data, indicating good generalizability of the approach: McNemar’s $\chi^2 = 101.68$, $df = 1$, p -value $< .0001$, Cohen’s $g = 0.47$ (large effect size).

7 Discussion

The results show that our data-driven approach substantially improves feedback performance, in particular the specificity of provided feedback. Comparing the baseline and updated datasets, we observe more instances of specific feedback, fewer instances of default feedback, and higher recall

and precision scores. These findings show that even limited rule enhancements motivated by user interaction logs yield notable gains. Thus, regarding **RQ 1**, we find that the authentic logs provided valuable insights into actual learner needs, which, in combination with a thorough analysis of the expected target answers, enabled targeted refinement of the rule set. The evaluation results on the testing data show that the improvements to the rules are stable and, even with unseen data, increase precision and recall of the feedback performance. This suggests a positive answer to **RQ 2**: our data-driven enhancement to rule-based feedback is likely to generalize well and perform similarly on unseen data.

Although the increase in precision and recall was higher on the unseen data, an unexpected outcome, further examination of the unseen dataset identified two causes. One is a higher percentage of incorrect question-specific feedback in baseline_U (3.68%) which allowed for more room for improvement relative to baseline_S (1.26%). A second reason is the simpler linguistic structures in the testing data due to changes to FeedBook’s exercise generation and exercise selection between the first and second data collection, which potentially allowed for better NLP parsing and therefore more accurate diagnoses by the feedback system. Based on these observations, we expect the updated system to perform similarly on unseen data if the aforementioned factors are controlled for or the testing data is sampled from the same dataset as the training data.

Furthermore, the results of our student answer investigations revealed a great variety of errors and misconceptions that are challenging to handle by a system or anticipate in general. We also observed a wider range of errors in full sentence answers, concluding that while our rule-based approach is precise and easily adjustable for shorter gap-fills, an approach using LLMs could be beneficial for longer input.

8 Conclusion and Outlook

In this paper, we evaluated and improved a rule-based feedback mechanism in an ITS using authentic learner interaction logs. Our analysis demonstrated that such real-world data provides valuable insights into actual learner difficulties and can meaningfully guide refinements of rule-based feedback. The targeted rule up-

Initial rules									
Feedback	QS		Spelling		DF		Total		
	✓	✗	✓	✗	✓	◆	✓	✗	✓+◆
Word	18	20	3	3	3	33	24	23	57
Phrase	32	1	33	19	7	144	72	20	216
Sentence	38	0	50	0	0	166	88	0	254
Total	88	21	86	22	10	343	184	43	527
						(60.18%)	(32.28%)	(7.54%)	(92.46%)
Updated rules									
Word	40	0	3	3	3	31	46	3	77
Phrase	70	1	33	19	7	106	110	20	216
Sentence	92	2	50	0	0	110	142	2	252
Total	202	3	86	22	10	247	298	25	545
						(43.33%)	(52.28%)	(4.39%)	(95.61%)

Table 7: Results for initial and updated rules on **testing data**: Raw numbers of provided feedback annotated as optimal (✓), incorrect (✗) and unspecific (◆) per subset (word, phrase, and sentence).

Initial rule set	Updated rule set		Total
	✓	✗ + ◆	
✓	183	4	187
✗ + ◆	115	268	383
Total	298	272	570

Table 8: Contingency table for the testing data.

dates led to a reduction in default and incorrect feedback, while increasing both precision and recall. These improvements generalized well to unseen data, confirming that authentic logs help strengthen rule-based systems while retaining pedagogical control over generated feedback.

Although we focused on improving an existing rule set, the proposed methodological flow has wider potential. Beyond rule-based feedback, future work could apply the same data-driven evaluation approach to LLM-based feedback generation or explore hybrid models that leverage both expert knowledge and data-driven flexibility. Training LLMs on learner data to produce new rules, or applying the approach to other exercise types, linguistic structures, or learner populations, represents promising avenues for future work.

Our work also lays the foundation for a direct comparison between the refined rule-based approach and an LLM-based one. While we argue that rule-based feedback may be particularly advantageous for gap-filling grammar exercises, especially due to its transparency and reliability, this assumption has yet to be empirically tested.

Comparing both approaches within the same evaluation framework will allow a better understanding of their respective strengths and the contexts in which each approach is most suitable.

Finally, although our analysis focused on internal evaluation, authentic data also offers valuable insights for external evaluation by indicating how many learners would benefit from specific refinements. Overall, our findings demonstrate that leveraging real learner data enables systematic, generalizable improvements to feedback mechanisms and supports the development of future feedback technologies in CALL.

9 Limitations

Data. Due to differences in data collection, the data from two periods varied significantly, which might influence the results but was not controlled for in this experiment. The most relevant differences are as follows:

- In terms of feedback, in the first data collection metalinguistic information in the feedback popups was replaced with the default message for some students while others received full access to the metalinguistic information during a field trial. There were no such treatment variations in this regard during the second data collection.
- All the exercises for the first data collection were manually crafted and implemented in the system. Later, however, only the exercise specifications such as vocabulary, grammar

context and others, were designed manually, while the exercises themselves were generated automatically based on these specifications (Heck et al., 2022). As we discuss in Section 7, this may result in more exercises with similar syntactic structures in the testing data and therefore, cause better feedback performance.

- In the first data collection, students had access to all the exercises at once and could select exercises from any difficulty level. In the second data collection, however, an algorithm assigned the most appropriate difficulty level to each student according to their performance (Colling et al., 2023). Such macro-adaptive selection could potentially lead to more students completing easier exercises and, therefore, to more data with simpler linguistic structures in the testing dataset.
- In real-life application, some of the student answer processing (e.g., removal of extra white space before punctuation, replacing characters of one encoding by the ones in another) currently happens in the front-end but did not take place when the training data was collected. Thus, some student answers that the system treated as correct in the second data collection were still included in the first data collection as malformed and contributed to the numbers with incorrect spelling or default feedback in our baseline.
- In contrast to the first data collection, the system from the second data collection does not consider capitalization a potential source of error in order to facilitate tablet and smartphone usage and to avoid over-punishing students on automatic device behavior, i.e. capitalizing initial characters in an input field.

Feedback mechanism. Most of the feedback rules rely on dependency parsing and annotations provided by NLP tools, such as UIMA. However, these tools are sometimes incorrect in their annotations, which can lead to unexpected results, e.g., default feedback for *do – does – did* confusion because of incorrect lemma value in the annotation provided by the UIMA Java SDK 2.9.0 version. We acknowledge that the latest version might yield better results and we encourage the use of a newer NLP pipeline in general.

Rules selection. In this paper, we selected rules to improve based on their performance in

the baseline_S, prioritizing rules that performed poorly with certain linguistic constructs related exclusively to interrogative sentences (e.g., question words or auxiliary verbs in the simple past and simple present tense). However, we acknowledge that another approach to rule selection, for instance, to triage learner errors according to how frequently they go undiagnosed, might have shown even greater gains, especially in recall.

Moreover, many student answers were not ill-formed, but well-formed questions that are incorrect due to the context, exercise instruction, and nature of the exercise itself. We acknowledge that the data units with such student answers had a certain impact on precision and, particularly, recall scores (most of these cases received default feedback). However, the current analysis did not focus on feedback related to, for example, tense confusion or erroneous use of pronouns.

Increasing precision. Most of the incorrect feedback provided in the training baseline was spelling feedback. It was erroneously given to student answers with non-question grammatical issues, answers that contained context (text around the target answer but not part of it), nonsense answers, and only in a few cases to answers with question-specific errors. Increasing the feedback precision by accounting for all the observed issues was out of the scope of this paper, but is a promising venue for further enhancement of the feedback algorithm.

Acknowledgments

This work is based on data from the Interact4School and AI2Teach projects that were financially supported by the German Ministry of Education and Science (BMBF DLR, funding code: 01JD1905A) and by the AIM (Akademie für Innovative Bildung und Management Heilbronn-Franken gemeinnützige GmbH), respectively.

We thank Walid El Hefny for his help with data extraction as well as Daniela Verratti Suoto and Sonja Gelzenleuchter for their thorough annotation efforts. We also thank Matthew Pattemore for his support in revising and polishing the paper and Kate Belcher for insightful discussions and contributions to shaping the final version. Last but not least, we thank the reviewers for their thoughtful comments and constructive feedback.

References

- Alan Agresti. 2012. *Categorical data analysis*, volume 792. John Wiley & Sons.
- Luiz Amaral, Detmar Meurers, and Ramon Ziai. 2011. Analyzing learner language: towards a flexible natural language processing architecture for intelligent language tutors. *Computer Assisted Language Learning*, 24(1):1–16.
- David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Journal of AI*, 7(1):52–62.
- Seyyed Kazem Banihashem, Omid Noroozi, Stan van Ginkel, Leah P Macfadyen, and Harm JA Biemans. 2022. A systematic review of the role of learning analytics in enhancing feedback practices in higher education. *Educational Research Review*, page 100489.
- Florian Berens, Katharina Wendebourg, Mareike Kholin, Leona Colling, Julia Schmidt-Peterson, Manuel Hopp, Tanja Heck, Stephen Bodnar, Walid El Hefny, Florian Nuxoll, Christoph Deeg, Katja Krey, Josef Schrader, Hannes Schröter, Benjamin Nagengast, Detmar Meurers, and Ulrich Trautwein. 2024. AI2Teach: Effectiveness of a teacher training on technology-supported teaching and learning. Registry ID: 20365.1v1. Pre-registration of the study design.
- Dorina Bór. 2023. Improving the user experience of an intelligent tutoring system for proving mathematical statements: Master thesis.
- 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.
- Anderson Pinheiro Cavalcanti, Arthur Barbosa, Ruan Carvalho, Fred Freitas, Yi-Shan Tsai, Dragan Gašević, and Rafael Ferreira Mello. 2021. Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*, 2:100027.
- Jinho D Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 363–367.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
- Leona Colling, Tanja Heck, and Detmar Meurers. 2023. Reconciling Adaptivity and Task Orientation in the Student Dashboard of an Intelligent Language Tutoring System. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 288–299.
- Wei Dai, Jionghao Lin, Hua Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gašević, and Guanliang Chen. 2023. Can large language models provide feedback to students? A case study on ChatGPT. In *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 323–325. IEEE.
- Jasper Degraeuwe and Patrick Goethals. 2024. Leading by example: The use of generative artificial intelligence to create pedagogically suitable example sentences. In *Swedish Language Technology Conference and NLP4CALL*, pages 33–48.
- Robert DeKeyser. 2007. Skill acquisition theory. In Bill VanPatten and Jessica Williams, editors, *Theories in Second Language Acquisition: An introduction*, volume 97113. Routledge.
- Bingyu Dong, Jie Bai, Tao Xu, and Yun Zhou. 2024. Large language models in education: A systematic review. In *2024 6th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 131–134. IEEE.
- Rod Ellis. 2006. Current issues in the teaching of grammar: An SLA perspective. *TESOL quarterly*, 40(1):83–107.
- Emrah Emirtekin. 2025. Large Language Model-Powered Automated Assessment: A Systematic Review. *Applied Sciences*, 15(10):5683.
- Ali Kürşat Erümit, İsmail Çetin, Mehmet Kokoç, Temel Kösa, Vasif Nabiyeve, and Emine Selin Aygün. 2019. Designing a usability assessment process for adaptive intelligent tutoring systems: A case study. *Turkish Online Journal of Qualitative Inquiry*, 10(1):141–179.
- Ali Mohammad Fazilatfar, Mohadeseh Sedghi, and Mohammad Javad Ali Beigi. 2024. An attempt to improve grammatical structure retention: Impact of computer-mediated feedback on iranian EFL learners. *Journal of Studies in Language Learning and Teaching*, 1(2):289–306.
- Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. *arXiv preprint arXiv:2305.15676*.
- David Ferrucci, Adam Lally, Karin Verspoor, and Eric Nyberg. 2009. Unstructured Information Management Architecture (UIMA) Version 1.0. OASIS Standard.
- Qing-Ke Fu, Di Zou, Haoran Xie, and Gary Cheng. 2022. A review of AWE feedback: types, learning outcomes, and implications. *Computer Assisted Language Learning*, 37(1-2):179–221.

- Jim Greer and Mary Mark. 2016. [Evaluation methods for intelligent tutoring systems revisited](#). *International Journal of Artificial Intelligence in Education*, 26:387–392.
- Tanja Heck, Detmar Meurers, and Florian Nuxoll. 2022. [Automatic exercise generation to support macro-adaptivity in intelligent language tutoring systems](#). *Intelligent CALL, granular systems and learner data: short papers from EUROCALL 2022*, page 162.
- Trude Heift. 2010. [Prompting in CALL: A longitudinal study of learner uptake](#). *Modern Language Journal*, 94(2):198–216.
- Trude Heift. 2013. [Multiple learner errors and meaningful feedback](#). *CALICO Journal*, 20(3):533–548.
- Jennifer Hill and Rahul Simha. 2016. [Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams](#). In *Proceedings of the 11th workshop on innovative use of NLP for building educational applications*, pages 23–30.
- Andrea Horbach, Ronja Laarmann-Quante, Lucas Liebenow, Thorben Jansen, Stefan Keller, Jennifer Meyer, Torsten Zesch, and Johanna Fleckenstein. 2022. [Bringing automatic scoring into the classroom—measuring the impact of automated analytic feedback on student writing performance](#). In *Swedish Language Technology Conference and NLP4CALL*, pages 72–83.
- Xinyi Huang, Di Zou, Gary Cheng, Xieling Chen, and Haoran Xie. 2023. [Trends, research issues and applications of artificial intelligence in language education](#). *Educational Technology & Society*, 26(1):112–131.
- Choi Inn-Chull. 2019. [Exploring the potential of a computerized corrective feedback system based on a process-oriented qualitative error analysis](#). *J Eng Teach Movie Media*, 20(1):89–117.
- Timur Jaganov, John Blake, Julián Villegas, and Nicholas Carr. 2025. [Large Language Model-Driven Dynamic Assessment of Grammatical Accuracy in English Language Learner Writing](#). *arXiv preprint arXiv:2505.00931*.
- Abdulkadir Karaci, Zeynep Piri, Halil İbrahim Akyüz, and Göksal Bilgiçci. 2018. [Student perceptions of an intelligent tutoring system: A technology acceptance model perspective](#). *Online Submission*, 182(22):31–36.
- Anisia Katinskaia, Sardana Ivanova, and Roman Yangerber. 2019. [Multiple admissibility in language learning: Judging grammaticality using unlabeled data](#). In *Workshop on Balto-Slavic Natural Language Processing*, pages 12–22.
- Semere Kiros Bitew, Johannes Deleu, A Seza Doğruöz, Chris Develder, and Thomas Demeester. 2023. [Learning from Partially Annotated Data: Example-aware Creation of Gap-filling Exercises for Language Learning](#). *arXiv e-prints*, pages arXiv–2306.
- Fabienne M Van der Kleij, Remco CW Feskens, and Theo JHM 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.
- Susanne Knoop and Sabrina Wilske. 2013. [WordGap - Automatic Generation of Gap-Filling Vocabulary Exercises for Mobile Learning](#). In *Proceedings of Second Workshop NLP Computer-Assisted Language Learning*, pages 39–47.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Large Language Models Are State-of-the-Art Evaluator for Grammatical Error Correction](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77. Association for Computational Linguistics.
- Kenneth R. Koedinger, Emma Brunskill, Ryan S.J.d. Baker, Elizabeth A. McLaughlin, and John Stamper. 2013. [New potentials for data-driven intelligent tutoring system development and optimization](#). *AI Magazine*, 34(3):27–41.
- Ehsan Latif, Vincent Liu, and Xiaoming Zhai. 2026. [A systematic review of intelligent and robot tutoring systems: evolution, pedagogical design, and AI-driven classification](#). *Smart Learning Environments*, 13(1):1.
- Elizabeth Lavolette, Charlene Polio, and Jimin Kahng. 2015. [The accuracy of computer-assisted feedback and students’ responses to it](#). *Language, Learning & Technology*, 19(2).
- Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. [Automated grammatical error detection for language learners](#). Morgan & Claypool Publishers.
- Shawn Loewen. 2013. [The role of feedback](#). In *The Routledge handbook of second language acquisition*, chapter 3, pages 24–40. Routledge.
- Alison Mackey. 1999. [INPUT, INTERACTION, AND SECOND LANGUAGE DEVELOPMENT: An Empirical Study of Question Formation in ESL](#). *Studies in Second Language Acquisition*, 21(4):557–587.
- Ahmad Marouf, Rami Al-Dahdooh, Mahmoud Jamal Abu Ghali, Ali Osama Mahdi, Basem S Abunasser, and Samy S Abu-Naser. 2024. [Enhancing education with artificial intelligence: The role of intelligent tutoring systems](#). *International Journal of Engineering and Information Systems (IJEAIS)*, pages 10–16.

- Detmar Meurers, Kordula De Kuthy, Florian Nuxoll, Björn Rudzewitz, and Ramon Ziai. 2019. [Scaling up intervention studies to investigate real-life foreign language learning in school](#). *Annual Review of Applied Linguistics*, 39:161–188.
- Guido Minnen, John Carroll, and Darren Pearce. 2001. [Applied morphological processing of English](#). *Natural Language Engineering*, 7(3):207–233.
- Elham Mousavinasab, Nahid Zarifsanaiey, Sharareh R. Niakan Kalhori, Mahnaz Rakhshan, Leila Keikha, and Marjan Ghazi Saeedi. 2021. [Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods](#). *Interactive Learning Environments*, 29(1):142–163.
- 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.
- 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.
- Ryo Nagata, Kentaro Inui, and Shin’ichiro Ishikawa. 2020. [Creating corpora for research in feedback comment generation](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhashnyi, Artem Chernodub, Oleksandr Kornienko, and Igor Samokhin. 2024. [Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33. Association for Computational Linguistics.
- Subhadarshi Panda, Frank Palma Gomez, Michael Flor, and Alla Rozovskaya. 2022. [Automatic generation of distractors for fill-in-the-blank exercises with round-trip neural machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 391–401.
- Cora Parrisius, Ines Pieronczyk, Carolyn Blume, Katharina Wendebourg, Diana Pili-Moss, Mirjam Assmann, Sabine Beilharz, Stephen Bodnar, Leona Colling, Heiko Holz, et al. 2022. [Using an intelligent tutoring system within a task-based learning approach in English as a foreign language classes to foster motivation and learning outcome \(Interact4School\): Pre-registration of the study design](#).
- Kenneth A Petersen. 2010. [Implicit corrective feedback in computer-guided interaction: Does mode matter?](#) Georgetown University.
- Manfred Pienemann, Malcolm Johnston, and Geoff Brindley. 1988. [Constructing an acquisition-based procedure for second language assessment](#). *Studies in second language acquisition*, 10(2):217–243.
- Cristobal Romero and Sebastian Ventura. 2013. [Data mining in education](#). *Wiley Interdisciplinary Reviews: Data mining and knowledge discovery*, 3(1):12–27.
- Cristobal Romero and Sebastian Ventura. 2020. [Educational data mining and learning analytics: An updated survey](#). *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 10(3):e1355.
- Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, and Detmar Meurers. 2017. [Developing a web-based workbook for English supporting the interaction of students and teachers](#). In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 36–46.
- Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. 2018. [Generating feedback for English foreign language exercises](#). In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 127–136.
- Simón Ruiz, Patrick Rebuschat, and Detmar Meurers. 2023. [Supporting individualized practice through intelligent CALL](#). In *Practice and automatization in second language research*, pages 119–143. Routledge.
- Helmut Schmid. 2005. [A programming language for finite state transducers](#). In *Finite-State Methods and Natural Language Processing (FSMNL)*, volume 4002 of *Lecture Notes in Computer Science*, pages 308–309. Springer.
- Julika Siemer and Marios C Angelides. 1998. [A comprehensive method for the evaluation of complete intelligent tutoring systems](#). *Decision support systems*, 22(1):85–102.
- Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. [GEE! Grammar Error Explanation with Large Language Models](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM prompting strategies for joint essay scoring and feedback generation](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.

- John Stamper, Ruiwei Xiao, and Xinying Hou. 2024. [Enhancing LLM-based feedback: Insights from intelligent tutoring systems and the learning sciences](#). In *International Conference on Artificial Intelligence in Education*, pages 32–43. Springer.
- Aru Ukenova and Gulmira Bekmanova. 2023. [A review of intelligent interactive learning methods](#). *Frontiers in Computer Science*, 5.
- Kurt VanLehn. 2011. [The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems](#). *Educational psychologist*, 46(4):197–221.
- Elena Volodina, Christopher Bryant, Andrew Caines, Orphée De Clercq, Jennifer-Carmen Frey, Elizaveta Ershova, Alexandr Rosen, and Olga Vinogradova. 2023. [MultiGED-2023 shared task at NLP4CALL: Multilingual Grammatical Error Detection](#). In *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning*, pages 1–16.
- 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.
- Lu Yang and Rui Li. 2024. [ChatGPT for L2 learning: Current status and implications](#). *System*, 124:103351.
- Jingheng Ye, Shang Qin, Yinghui Li, Hai-Tao Zheng, Shen Wang, and Qingsong Wen. 2025. [Corrections Meet Explanations: A Unified Framework for Explainable Grammatical Error Correction](#). *arXiv preprint arXiv:2502.15261*.
- Chak Yan Yeung, John SY Lee, and Benjamin K Tsou. 2019. [Difficulty-aware distractor generation for gap-fill items](#). In *Proceedings of the 17th annual workshop of the Australasian language technology association*, pages 159–164.
- Nana Yoshimi, Tomoyuki Kajiwara, Satoru Uchida, Yuki Arase, and Takashi Ninomiya. 2023. [Distractor generation for fill-in-the-blank exercises by question type](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 276–281.
- M Zerkouk, M Mihoubi, and B Chikhaoui. 2025. [A comprehensive review of AI-based intelligent tutoring systems: Applications and challenges](#). *arXiv preprint arXiv:2507.18882*.

A Rule Change Example

Figures 3 and 4 provide snippets of simplified pseudocode to illustrate a change from the token-position approach to the syntactic dependency one

in a rule. The example shows *Question Replacement* rule that searches for a question word in a target answer (TA) and, if it identifies one, replaces the question word with a different one from a provided list of question words.

Figure 5 shows the different components and the application flow of a feedback rule in the Feed-Book.

```
[...]
questionWords = "Whose", "Which", "Where",
"Whom", "What", "When",
"Why", "Who", "How"
[...]
isRuleApplicable(Configuration of TA):
    get a stringTA from Configuration
    if stringTA is null
        return false
    split stringTA by white space
    get the firstToken from stringTA
    if firstToken in questionWords
        if lastToken is "?"
            return true
        else return false
    else return false

applyRule(annotatedTA, indexTA):
    get the firstToken from annotatedTA
    replace firstToken with questionWord
    set correct value of firstToken
    set incorrect value of firstToken
    return extended annotatedTA copy
[...]
```

Figure 3: *Question Word Replacement* rule with token-position approach (simplified pseudo-code)

B Sampling Strategy for Unseen Data

As there was more data collected in the second period we did not take the entire dataset for evaluation but instead selected representative samples. We simulated one iteration of a 10-fold validation for each subset to create the testing (unseen) data.

Sampling for each subset contained two steps: 1) Random sample of exercises; 2) Stratified random sample of attempt data units from sampled exercises.

For step 1, to match the same number of exercises present in the training dataset, and thus have a similar distribution of different exercise contexts, we randomly selected exercises from the

```

[...]
isRuleApplicable(Configuration of TA):
    for every dependency in TA span:
        if dependencyType is "advmod" and
           questionWords contains dependent
            return true
    return false

applyRule(annotatedTA, indexTA):
    for every dependency in TA span:
        if dependencyType is "advmod" and
           questionWords contains dependent
            replace dependent
            set correct value of dependent
            set incorrect value of dependent
    return extended annotatedTA copy
[...]

```

Figure 4: *Question Word Replacement* rule with syntactic dependency approach (simplified pseudo-code)

second period data to be included in the testing dataset. For Word, the training dataset contained 15 exercises. However, due to the adaptive exercise selection feature introduced between data collections (see Section 9), there were fewer instances with one-word target answers. Thus, all eight available exercises were included for the Word subset.

In step 2, a stratified sampling approach on the remaining data was conducted with the different exercises as strata. During the sampling, one constraint was implemented: Once an attempt from a student was included in the sample, the entire sequence of attempts for this student and this particular exercise was included to capture entire interaction sequences as present in the training dataset. For Phrase and Sentence, this resulted in minimally more data units included in the sample than calculated according to the 90:10 ratio.

C Feedback Messages

Table 9 provides the set of teacher-crafted feedback messages for question-specific learner errors including such general messages as default, spelling, and capitalization.

Table 10 provides the set of feedback messages for the rules added according to the learner needs identified during profiling of training data.

D Inter Annotator Agreement

Table 11 shows the inter-annotator agreement results for the two independent raters.

E Gap Filling Exercises in FeedBook

Figure 6 shows examples of semi-open gap filling grammar exercises of input types “word”, “phrase”, and “sentence” for learning target “Questions” in FeedBook.

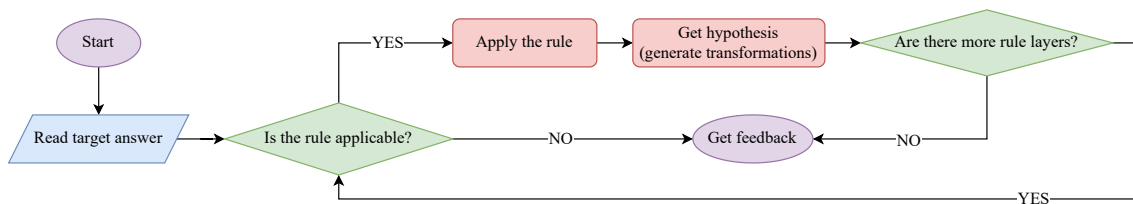


Figure 5: Feedback generation approach in the FeedBook system. Each rule consists of three methods: 1) Check whether the rule is applicable to the TA, i.e. relevant linguistic information is present in the TA to perform a specific transformation (e.g., *Question Word Replacement* is not applicable for “Did you sleep well?” but applicable for “Where did you see him?”); 2) Apply transformations to the TA in order to generate ill-formed and well-formed incorrect variations, and 3) get the error diagnosis for future feedback mapping.

Error	Message
Incorrect question word	Use another question word.
Missing question mark	You’ve forgotten to use the ‘?’.
Missing question word	You’ve forgotten to use a question word.
Missing auxiliary word	You’ve forgotten an auxiliary verb (‘Hilfsverb’, e.g. ‘do’/‘did’).
Incorrect auxiliary word	That’s not the correct auxiliary verb (‘Hilfsverb’).
Auxiliary and main verb are both in past tense	In questions with ‘did’ and a main verb (e.g., “Did you go?”), the main verb needs to be in the infinitive.
Incorrect agreement of auxiliary verb ‘do’	That’s not the correct form of an auxiliary verb (‘Hilfsverb’).
Auxiliary and main verb are both in third person	In questions with ‘does’/‘do’ and a main verb (e.g., “Where does he live?”), the main verb needs to be in the infinitive.
Spelling error	The following words in your answer seem to contain spelling errors: [the word from the student answer with a spelling mistake].
Capitalization error	There is a problem with capitalization in your answer.
Default feedback	This is not what I am expecting - please try again.

Table 9: Teacher-crafted feedback messages for the initial rules under analysis.

Error	Message
Main verb in past tense, auxiliary verb in infinitive	There seems to be a mistake in inflection. Look at this example: “What did you eat for breakfast?” and try to use the same structure in your question. Pay attention to the form of the auxiliary verb (‘Hilfsverb’, e.g. ‘did’) and the main verb.
Missing main verb	You forgot the main verb.
Main verb ‘be’ replaced by an auxiliary verb	You do not need an auxiliary verb (‘Hilfsverb’) here, because the main verb is a form of the verb ‘to be’. Remember to use the correct form of the verb ‘to be’ here.
Incorrect agreement of the verb ‘be’ as an auxiliary or main verb	You should use a different form of the verb ‘to be’.
Incorrect word order	The words in your answer are in the wrong order.

Table 10: Feedback messages for the added rules and a message for the additional word order check.

Read the answers and complete the questions.

_____ you know how to swim? → No, I don't know how to swim.

_____ was Ben wet? → Ben was wet because it rained.

_____ they have some water? → Yes, they may have some water.

_____ Paul watch the game every day? → Yes, Paul watches the game every day.

_____ the twins home? → No, the twins are not home.

Cancel

(a) Gap filling exercise with single word input.

Read the answers and complete the questions.

_____ Ben wet? → Ben was wet because it rained.

_____ Emma in Italy? → Emma was in Italy in June.

_____ the books? → The books are on the shelf.

_____ the cool musician? → The cool musician is Jimi Hendrix.

_____ his food? → His food was delicious.

Submit Exercise

(b) Gap filling exercise with phrase input.

Read the answers. Which questions do you need to ask if you want to find out about the underlined information?

1. Answer: No, it was not hot.

2. Answer: I can help her.

3. Answer: They finished their project yesterday.

4. Answer: No, I don't clean my room myself.

5. Answer: No, the book was not fun to read.

Submit Exercise

(c) Gap filling exercise with full sentence input.

Figure 6: Examples of semi-open gap filling grammar exercises for learning target “Questions” in FeedBook.

baseline_S (Total: 5076)	correct-specific	correct-non-specific	incorrect	unspecific
correct-specific	1676	0	12	0
correct-non-specific	0	69	0	0
incorrect	67	0	215	0
unspecific	0	1	0	3036
Inter-annotator agreement, Cohen's Kappa = 0.97 (Cohen, 1960)				
update_S (Total: 5076)	correct-specific	correct-non-specific	incorrect	unspecific
correct-specific	2157	0	14	0
correct-non-specific	0	56	0	0
incorrect	22	0	235	0
unspecific	0	14	0	2578
Inter-annotator agreement, Cohen's Kappa = 0.98				
baseline_U (Total: 570)	correct-specific	correct-non-specific	incorrect	unspecific
correct-specific	169	0	7	0
correct-non-specific	0	7	0	3
incorrect	0	0	38	2
unspecific	0	1	0	340
Inter-annotator agreement, Cohen's Kappa = 0.96				
update_U (Total: 570)	correct-specific	correct-non-specific	incorrect	unspecific
correct-specific	286	0	8	0
correct-non-specific	0	8	0	2
incorrect	1	0	18	0
unspecific	0	0	0	248
Inter-annotator agreement, Cohen's Kappa = 0.97				

Table 11: Inter Annotator Agreement.

A Bigger Catch: Fine-Grained Curriculum Standards Alignment on the MathFish Benchmark

Xinman Liu, Mayank Sharma, Teah Shi

Stanford University

{xinman, masharma, teah2001}@stanford.edu

Abstract

Most existing math benchmarks for LLMs focus on evaluating correct solutions. In educational settings, however, it is equally important to understand whether LLMs understand the *pedagogical intent* behind math problems, beyond producing the right solution. Tagging curriculum standards is challenging for the same reason: distinguishing fine-grained standards requires understanding subtle pedagogical distinctions. In this paper, we use the MathFish benchmark, which frames alignment as multi-label prediction over 385 Common Core State Standards, to evaluate a three-stage pipeline inspired by baseline failure modes in retrieval and structural reasoning: curriculum-informed hard negatives (M1), a cross-encoder re-ranker (M2), and a ReAct agent paired with an LLM-as-a-judge critic (M3). We additionally evaluate a training-free alternative (A1) combining hybrid sparse-dense retrieval with curriculum graph reranking. M3 achieves 31.3% exact match, roughly $6.5\times$ the three-shot GPT-4-Turbo baseline (Lucy et al., 2024). Error analysis shows that even with these improvements, the pipeline struggles with missing predictions, grade-level misalignment, and sibling confusion, reinforcing that precise curriculum alignment is a fundamentally hard problem in educational NLP.

1 Introduction

Most existing benchmarks for mathematical reasoning in language models ask one question above all else. Can the model get the right answer? Datasets such as GSM8k (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) test whether a model can solve a problem, but they rarely examine what mathematical competencies that problem actually exercises. In actual K-12 education, though, the categorization of math content matters more than it might seem. Professional curriculum reviewers spend months mapping published math problems to fine-grained pedagogical standards (Lucy

et al., 2024). As LMs are increasingly deployed in classrooms for applications such as generating assessments or tutoring dialogues (Shi et al., 2026), it becomes important to understand whether they actually grasp the pedagogical structure behind math questions or just answer them correctly. The MathFish benchmark (Lucy et al., 2024) was introduced to evaluate exactly this, framing curriculum alignment as a multi-label prediction task over 385 math standards derived from the Common Core State Standards for Mathematics (Common Core State Standards Initiative, 2010). The standards follow the hierarchical structure of CCSSM and are commonly operationalized using resources from Achieve the Core (Student Achievement Partners, 2024). Each math problem is annotated with the standards it assesses, requiring models to predict the relevant labels from the full taxonomy.

This task turns out to be surprisingly **hard**. Many standards share enough surface vocabulary that telling them apart demands genuine understanding of pedagogical intent. When GPT-4 is evaluated with three-shot prompting and no hierarchy hints, exact-match accuracy reaches only 4.8% (Lucy et al., 2024). Most incorrect predictions fall on structurally nearby standards (siblings). In this paper, we present a three-stage pipeline where each stage is motivated by a specific failure mode of the previous one: off-the-shelf embeddings fail to separate pedagogically similar standards, so we train a contrastive bi-encoder with curriculum-informed hard negatives (M1); independent encoding misses fine-grained problem-standard interactions, so we add a cross-encoder re-ranker (M2); and threshold-based selection cannot reason about pedagogical intent, so we attach a ReAct agent with an LLM-as-a-judge critic (M3). We additionally evaluate a training-free alternative (A1) that replaces the learned retrieval front-end with hybrid sparse-dense retrieval and curriculum graph reranking, testing whether structured graph knowledge can substitute

for fine-tuning. Each stage is described in detail in Section 3.

Using M3, we achieve **31.3% exact match, roughly $6.5\times$ the GPT-4 baseline** (Lucy et al., 2024), and discuss the roles of retrieval, reranking, and LLM reasoning in achieving precise alignment, as well as further challenges in performing this task. Our code is available at <https://github.com/yo-lxmmm/a-bigger-catch-mathfish>.

2 Related Work

Early research framed standards alignment as a multi-label text categorization problem. Using manually aligned benchmarks as training data, SVMs with bag-of-words features were applied to map curricular content to standards, showing feasibility but highlighting challenges from short standard texts and substantial lexical overlap between unrelated concepts (Yilmazel et al., 2007). Fine-tuning LMs such as BERT on task-specific educational text substantially improves classification accuracy (Shen et al., 2021), and sentence embedding models have been used to align educational resources with skill taxonomies in a shared embedding space (Li et al., 2024). However, off-the-shelf embeddings remain unreliable for math specifically: cosine similarities between different skill category embeddings often exceed 0.88 because problems share a homogeneous vocabulary of numbers, operations, and geometric terms (Xu et al., 2025). Together, these studies motivate task-specific representation learning as a prerequisite for reliable curriculum retrieval, which is the role M1 is designed to fill.

2.1 Retrieval and Reranking for Educational Tagging

TagRec (V et al., 2021) and TagRec++ (Viswanathan et al., 2022) are the closest prior applications of dense retrieval to education, framing question-to-taxonomy tagging as a similarity-based retrieval problem and showing that retrieval-based models outperform flat multi-class classifiers on hierarchical educational taxonomies. Standard contrastive objectives treat closely related labels as equally negative as completely unrelated ones; hierarchy-aware contrastive learning approaches such as Use All The Labels (Zhang et al., 2022) incorporate label structure directly into the loss. Our hard negative sampling strategy in M1 follows this intuition by

oversampling standards nearby in the ATC graph, reflecting where real confusion occurs. However, bi-encoders encode problems and standards independently, limiting fine-grained interaction between them. Two-stage retrieve-then-rerank pipelines address this limitation: BEIR (Thakur et al., 2021) confirms that cross-encoder re-ranking achieves strong performance across diverse domains, motivating our M2 stage.

2.2 Agentic Reasoning and Graph-Augmented Retrieval

The ReAct framework (Yao et al., 2023) allows an LLM to interleave reasoning steps with tool calls, grounding decisions in external knowledge rather than parametric memory alone. Our M3 stage pairs a ReAct agent with an LLM-as-a-judge critic (Zheng et al., 2023) that prunes the candidate set before committing to a final prediction. As a training-free alternative, A1 replaces the learned retrieval front-end with a hybrid sparse-dense retriever (Luan et al., 2021) combining BM25 (Robertson and Zaragoza, 2009) with dense embeddings, reranked via GraphRAG (Edge et al., 2025). Recent work in educational retrieval (Jain et al., 2025) shows that graph-augmented retrieval generates more relationally coherent responses to curriculum-level queries than flat vector-based RAG, supporting the graph reranking design in A1.

2.3 The MathFish Benchmark

Our work builds directly on MathFish (Lucy et al., 2024), which frames curriculum alignment as multi-label prediction over 385 CCSS standards organized in a four-level hierarchy (grade, domain, cluster, standard) with 1,040 conceptual links in the Achieve the Core coherence map. Even GPT-4 with three-shot prompting achieves only 0.048 exact-match accuracy, and errors are structured rather than random: performance degrades as distractor standards become conceptually closer to the correct one, particularly within shared domains or ATC graph neighbors. This structured error pattern directly motivates our pipeline design—curriculum-informed hard negatives in M1 target the confusion clusters identified in MathFish’s error analysis, and the ReAct agent in M3 is equipped with tools to explicitly traverse the ATC graph during reasoning.

3 Approach

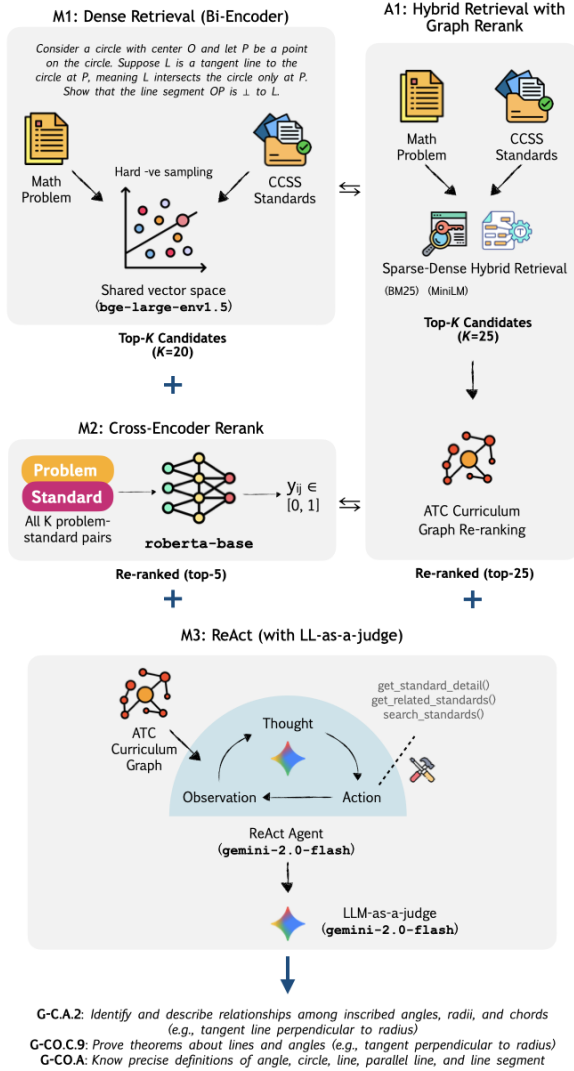


Figure 1: Overview of our pipeline. M1 (top left) trains a contrastive bi-encoder with curriculum-informed hard negatives to retrieve top-20 candidate standards from a shared embedding space. M2 (middle left) re-ranks the top-20 candidates using a roberta-base cross-encoder that jointly scores each problem-standard pair. A1 (top right) is a training-free alternative that replaces M1 and M2 with sparsedense hybrid retrieval followed by ATC curriculum graph reranking. All pipelines pass their top candidates to M3 (bottom), a ReAct agent powered by gemini-2.0-flash that iteratively reasons over standard descriptions and curriculum graph neighbors, followed by an LLM-as-a-judge critic that prunes the final prediction set.

3.1 Contrastive Bi-Encoder with Hard Negatives (M1)

For our first stage, we frame standard alignment as retrieval over 385 candidates without hierarchy hints, and adopt a bi-encoder architecture

for its ability to precompute all standard embeddings and retrieve efficiently. The encoder uses BAAI/bge-large-en-v1.5 (335M parameters) as a shared transformer backbone, with separate 256-dimensional linear projection heads for problems and standards. All embeddings are L2-normalized and scored by dot product. Given a problem embedding $\mathbf{q} = g_p(\text{enc}(x))$, a positive standard embedding $\mathbf{s}^+$, and n hard negatives $\{\mathbf{s}_j^-\}_{j=1}^n$, the model minimizes the InfoNCE contrastive loss

$$\mathcal{L}_{\text{bi}} = -\log \frac{\exp(\mathbf{q} \cdot \mathbf{s}^+ / \tau)}{\exp(\mathbf{q} \cdot \mathbf{s}^+ / \tau) + \sum_{j=1}^n \exp(\mathbf{q} \cdot \mathbf{s}_j^- / \tau)} \quad (1)$$

where τ is the temperature parameter controlling the sharpness of the similarity distribution. What distinguishes our approach is **curriculum-informed hard negative sampling**. MathFish’s error analysis reveals that model confusions follow predictable structural patterns along the ATC hierarchy. We therefore construct hard negatives that reflect where real confusion occurs. For each (problem, positive standard) pair, we sample $n = 8$ negatives from the curriculum graph following a fixed ratio of 40% siblings (same cluster), 30% conceptual neighbors (ATC coherence links), 20% grade-adjacent standards (± 1 grade), and 10% random, excluding all gold standards for that problem. This ratio was chosen to approximate the distribution of structured confusions reported in the MathFish error analysis (Lucy et al., 2024), where sibling and conceptual-neighbor errors account for the majority of misclassifications. At inference, each problem is encoded and scored against all 385 precomputed standard embeddings, returning the top- k candidates.

3.2 + Cross-Encoder Re-Ranking (M2)

Our second stage builds on the bi-encoder (Section 3.1) by adding a cross-encoder re-ranker that jointly processes each (problem, standard) pair through full bidirectional attention. While the bi-encoder retrieves candidates efficiently, it encodes problems and standards independently, limiting the depth of interaction between them. The cross-encoder addresses this, returning the top- n re-ranked candidates as the final prediction.

We use roberta-base (125M parameters) with a binary classification head that outputs a relevance score $\hat{y}_{ij} \in [0, 1]$ for each candidate pair (x_i, s_j) . For each of the bi-encoder’s top- k candidates ($k = 20$),

the problem text and standard description are concatenated into a single input sequence and passed through the full transformer. The model is trained with binary cross-entropy loss

$$\mathcal{L}_{\text{re}} = -\frac{1}{|P|} \sum_{(x_i, s_j) \in P} [y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij})] \quad (2)$$

where $y_{ij} \in \{0, 1\}$ indicates whether standard s_j is a gold alignment for problem x_i . Training pairs come from the trained bi-encoder’s top-20 predictions on the training set; gold standards that appear in this candidate list serve as positives, and the rest as negatives. At inference, the cross-encoder scores all 20 bi-encoder candidates per problem, re-ranks them by score, and returns the top- n ($n = 5$) as the final prediction. The two-stage pipeline separates the retrieval task, which must search efficiently across all 385 standards, from the re-ranking task, which can afford deeper cross-attention over a focused candidate set.

3.3 + ReAct with LLM-as-a-judge (M3)

Our third stage builds directly on the first two (Sections 3.1 and 3.2), reusing the trained bi-encoder and cross-encoder as the retrieval and reranking front-end, then attaching a ReAct-based LLM agent (Yao et al., 2023) and an LLM-as-a-judge critic (Zheng et al., 2023) to the cross-encoder’s top-5 output. This replaces threshold-based top- n selection with deliberative multi-step reasoning over the candidate set, addressing the limitation that the retrieve-then-rerank pipeline lacks the capacity to identify the precise gold set.

Agent. The cross-encoder’s top-5 candidates are passed as the initial candidate list to a ReAct agent (Yao et al., 2023) powered by gemini-2.0-flash-001 on Vertex AI (see Appendix A.1). The agent reasons over the candidate set through repeated THOUGHT $\rightarrow$ ACTION $\rightarrow$ OBSERVATION cycles, up to a maximum of 12 steps per problem. At each cycle, the agent assesses whether the current candidates sufficiently capture the problem’s mathematical intent before invoking one of three tools: `get_standard_detail` to retrieve a standard’s full description; `get_related_standards` to explore sibling and conceptual neighbors in the curriculum graph; or `search_standards` to issue a new lexical query when candidates appear misaligned. Each tool response is returned as an observation, informing the next reasoning step. Once the agent determines a final alignment, it ends with `Final Answer: <standard IDs or none>`.

Critic. An LLM-as-a-judge critic (Zheng et al., 2023) powered by the same model (gemini-2.0-flash-001) prunes the predicted standard set (see Appendix A.2). Given the problem and the agent’s predictions with their full descriptions, the critic returns the smallest subset that directly matches the mathematical skills being assessed, or none if no candidate clearly aligns. The final prediction is formed by intersecting the critic’s output with the agent’s original set, guarding against hallucinated identifiers.

3.4 Hybrid Retrieval with Graph Reranking and ReAct Agent (A1)

We also explore a training-free alternative (A1) which replaces the trained bi-encoder and cross-encoder with a training-free hybrid retriever and curriculum graph reranker while reusing the same ReAct agent and critic from Section 3.3, serving as a training-free comparison to the pipelines above.

Retrieval and reranking. Standards are indexed via a sparse-dense hybrid retriever (Luan et al., 2021) combining BM25 (Robertson and Zaragoza, 2009) and all-MiniLM-L6-v2, returning $k = 25$ candidates per problem. These are reranked by assigning each candidate a base score $\frac{1}{1 + \text{rank}}$ and a coherence bonus computed against the top-5 retrieved candidates as seeds. For each (candidate, seed) pair we apply a priority cascade consistent with the ATC hierarchy, with weights reflecting the relative strength of each proximity signal: +0.10 for a shared CCSSM cluster (stopping further seed comparisons), +0.05 for a shared domain code, or +0.03 for a shortest path of ≤ 2 hops on the undirected ATC coherence graph, with domain and graph bonuses accumulating across seeds. All 25 reranked candidates are then passed to the ReAct agent and critic, providing a broader pool than M3’s top-5. This architecture was selected after evaluating several retrieval variants on a 100-problem development subset (Appendix B), where graph reranking over hybrid-retrieved candidates outperformed sparse, hybrid, and graph-first retrieval-only alternatives.

4 Experiments

4.1 Dataset

We use MathFish (Lucy et al., 2024)¹, a dataset of approximately 21,776 math problems (13,065 train

¹<https://huggingface.co/datasets/allenai/mathfish>

/ 4,356 dev / 4,355 test) drawn from Illustrative Mathematics and Fishtank Learning curricula, each labeled with fine-grained Common Core State Standards (CCSS). The 385 standards are organized hierarchically across four levels (grade $\rightarrow$ domain $\rightarrow$ cluster $\rightarrow$ standard), with 1,040 conceptual connections between them and natural language descriptions provided for each. Using ATC standard metadata², we filter to problems containing at least one *Addressing* or *Alignment* label, yielding 5,956 training, 1,942 development, and 2,025 test problems. Following the evaluation protocol of the original MathFish paper (Lucy et al., 2024), which reports all results on its evaluation set rather than a held-out test split, we report on the development set throughout to enable direct comparison with the baseline. The bi-encoder and cross-encoder were trained exclusively on the filtered training set; the A1 variant selection (Appendix B) used a separate 100-problem subset of the development data, and no other hyperparameters were tuned on this set. An example problem and its associated standard labels is shown in Appendix C. Label distribution is skewed: across the 584 standards with at least one training example, n_s ranges from 1 to 139 (median 11, mean 15.0). 20.7% of standards have fewer than 5 training examples, and 44.0% have fewer than 10, reflecting the uneven coverage of K-12 curricula across open educational resources (see Appendix D for the full distribution).

4.2 Evaluation

We evaluate using the following metrics: *Exact Match* requires the predicted standards to exactly match gold; *Weak Accuracy* requires at least one overlap. *Micro/Macro F1* capture token-level and standard-level agreement. *Ret* and *Rerank* report recall@5 before and after reranking (cross-encoder for M2/M3, curriculum graph for A1). *GraphDist* measures the average minimum graph distance between predicted and gold standards in the ATC graph ($\downarrow$ better), and *SibConf* measures the fraction of errors landing on a same-cluster sibling ($\downarrow$ better). *Avg Pred* tracks the average number of predicted standards per problem.

4.3 Experimental Details

Bi-Encoder (M1). We initialize from BAAI/bge-large-en-v1.5 (335M parameters) and train on 5,956 problems using AdamW (learning rate $2 \times$

10^{-5} , batch size 16) for 15,000 gradient steps with temperature $\tau = 0.03$ and $n = 8$ curriculum-aware hard negatives per positive. The best checkpoint is selected by minimum validation loss. Training uses FP16 on a single NVIDIA H100 (80 GB) and completes in approximately 50 minutes.

Cross-Encoder Re-Ranker (M2). We fine-tune roberta-base (125M parameters) for 3 epochs using AdamW (learning rate 2×10^{-5} , batch size 32, max sequence length 256). Training pairs are constructed by running M1 on the training set with $k = 20$, yielding 119,120 scored pairs per epoch and 11,169 gradient steps total. Training completes in approximately 20 minutes on the same H100.

ReAct Agent and LLM-as-a-Judge (M3). M3 requires no training and operates at inference time. Both the ReAct agent and LLM-as-a-judge use gemini-2.0-flash-001.

Hybrid Retriever and Graph Reranker (A1). A1 also requires no training. BM25 sparse retrieval is combined with dense embeddings from all-MiniLM-L6-v2 and candidates are re-ranked using the ATC curriculum graph via GraphRAG before being passed to the same agentic reasoning mechanism as M3.

4.4 Results

Table 1 presents the main results for all pipeline stages and the training-free alternative against the results of three-shot GPT-4-Turbo under self-guided tree traversal reported in the original MathFish paper (Lucy et al., 2024) as the baseline; all evaluated on the same development set (Addressing/Alignment labels only).

The curriculum-aware contrastive bi-encoder (M1) achieves strong retrieval coverage with 20 standards per problem. Adding the cross-encoder re-ranker (M2) tightens the prediction set to 5 candidates and cuts graph distance from 4.24 to 2.42, confirming that cross-attention over concatenated problem-standard pairs captures alignment signals that independent embeddings miss. Both M1 and M2 reach 0.000 exact match despite strong weak accuracy, which is expected given that predicting 20 and 5 candidates respectively against a gold average of 1.47 makes exact match nearly impossible by construction—these are fixed-size prediction sets, not threshold-based ones. A fairer evaluation of reranking alone would apply a score threshold to M2’s cross-encoder outputs to produce a variable-size prediction set; we leave this to future work and note that the single-pass critic (28.4% EM)

²<https://huggingface.co/datasets/allenai/achieve-the-core>

Table 1: Main results on the full Addressing/Alignment-filtered development set (1,942 problems). — denotes metrics not reported.

Method	Accuracy				Recall@5		Graph Quality		Avg Pred*
	Exact	Weak Acc	Micro F1	Macro F1	Ret	Rerank	GraphDist↓	SibConf↓	
Baseline: Three-Shot GPT-4-Turbo [†]	0.048	0.502	—	—	—	—	1.90	—	3.05
M1: Biencoder	0.000	0.728	0.088	0.087	0.632	—	4.24	0.020	20.00
M2: Biencoder + Cross Rerank	0.000	0.688	0.266	0.259	0.632	0.688	2.42	0.060	5.00
M3: Biencoder + Cross Rerank + ReAct	0.313	0.589	0.455	0.478	0.632	0.688	0.93	0.109	1.37
A1: Hybrid + Graph Rerank + ReAct	0.275	0.541	0.401	0.432	0.271	0.311	1.047	0.100	1.438

[†]Three-shot GPT-4-Turbo under self-guided tree traversal, as reported in the original MathFish paper (Lucy et al., 2024); evaluated on the same development set (Addressing/Alignment labels only).

*Gold average = 1.47 for the A+A filtered set.

provides a practical estimate of what M2’s candidate pool can support with a downstream selector. This confirms that retrieval and reranking alone are insufficient without a mechanism to identify the precise gold set and reason over pedagogical intent.

As demonstrated in M3, attaching the ReAct agent and critic to M2’s top-5 candidates produces the largest gain in the ablation: exact match rises from 0.000 to 0.313, roughly 6.5 times the three-shot GPT-4-Turbo baseline (0.048) (Lucy et al., 2024). The agent’s multi-step reasoning (e.g., inspecting standard descriptions, exploring graph neighbors, and invoking the critic to prune over-predictions) converts a ranked list into a substantially more precise prediction set, reducing average predictions from 5.00 to 1.37 (close to the gold average of 1.47) and achieving the best Micro-F1 (0.455) and Macro-F1 (0.478) across all stages and the alternative. M3 also achieves the best graph distance at 0.93, suggesting that the cross-encoder’s structurally tightened candidate pool effectively guides the agent toward predictions that are close in the curriculum graph even when they do not exactly match gold.

To situate the contribution of fine-tuned retrieval, we compare M3 against A1. Despite passing a larger reranked candidate pool to the agent (top-25 vs. top-5), A1 achieves lower exact match (0.275 vs. 0.313) and higher graph distance (1.047 vs. 0.93). The performance gap is further reflected in recall at the top-5 candidate level: A1’s untuned hybrid retriever achieves retrieval recall of only 0.271, which improves marginally to 0.311 after graph reranking. In contrast, M3’s fine-tuned bi-encoder achieves retrieval recall of 0.632, preserved and improved to 0.688 after cross-encoder reranking. We note a confound in this comparison: A1 passes 25 candidates to the agent while M3 passes only 5, meaning the

gap could reflect lower retrieval recall, a larger and noisier candidate pool confusing the agent, or both. We leave a controlled ablation varying candidate pool size to future work, but note that A1’s retrieval recall at top-5 (0.311) is already substantially lower than M3’s (0.688), suggesting retrieval quality is the primary driver.

To further isolate the contribution of the ReAct agent’s iterative tool use from simply having a zero-shot pruning LLM call, we evaluated a zero-shot single-pass critic applied directly to M2’s top-5 using the same model and prompt as M3. This ablation achieves 28.4% exact match in 48 minutes (vs. 2.64 hours for M3), indicating that the trained retrieval pipeline (M1 and M2) accounts for the majority of the improvement over the prompting-only baseline, with the ReAct loop and tool use contributing a further +2.9 percentage points and meaningfully tighter graph distance (0.929 vs. 0.981) at the cost of additional inference time.

We note that M3 uses gemini-2.0-flash-001 while the baseline uses GPT-4-Turbo, confounding pipeline design with model choice. The A1 pipeline uses the same Gemini backbone as M3 yet achieves only 0.275 exact match vs. M3’s 0.313, and the single-pass critic ablation with Gemini already reaches 0.284 without any ReAct loop, together suggesting the retrieval pipeline contributes independently of the LLM backbone. The D* ablation (Appendix B) further confirms that model choice matters within our framework. We acknowledge that cleanly isolating the pipeline contribution would require holding the backbone constant, which we leave to future work.

In terms of computational cost, over the full 1,942-problem development set, M3 completes in 2.64 hours (4.89 s/problem), averaging 4.00 Gemini API calls per problem (approximately 3 ReAct steps plus 1 critic call); A1 is slightly slower at 5.38

s/problem (2.90 h total), averaging 4.54 calls per problem. Approximate API cost is \$2.00-\$2.50 for M3 and \$3.00-\$3.50 for A1 across the full development set, suggesting both pipelines are deployable at practical scale.

5 Analysis

To better understand where and why our systems succeed and fail, we analyze performance across curriculum strata, error patterns, and system behaviors beyond aggregate metrics.

Performance across curriculum strata. Examining the results for M3 (the best-performing configuration), performance varies substantially across grade levels and domains (Tables 5 and 6 in Appendix E). Exact match is strongest at early elementary grades (K-2: 0.27-0.30) and decreases toward upper elementary and secondary levels, reaching its lowest at grade 5 (0.12) and high school (0.13). Domain-level results show similar variance: Geometry (0.30) and Number & Operations in Base Ten (0.32) achieve the strongest performance, while Functions (0.09) and Fractions (0.10) the weakest. Such variance is consistent with the original MathFish paper which reports exact accuracy ranging from 0.151 (Counting & Cardinality) to 0.011 (Functions) for GPT-4 (Lucy et al., 2024), suggesting domain difficulty reflects intrinsic properties of the task. These grade- and domain-level patterns are reflected in the systematic failure modes we characterize below.

Label frequency and performance. To disentangle whether performance gaps reflect pedagogical difficulty or data sparsity, we examine the relationship between per-standard training frequency and M3 exact match (see Appendix D). At the problem level, we find a modest but significant correlation between training frequency of the rarest gold standard and exact match ($\rho = 0.234$, $p < 10^{-24}$): problems whose gold standards appear rarely in training (Q1, $n_s \leq 10$) achieve 14.8% EM, compared to 44.9% for problems with frequently-occurring standards (Q4, $n_s \geq 31$). However, when examined at the per-standard level directly, correlating each standard’s training frequency with its conditional exact-match rate across development problems, the relationship disappears entirely ($\rho = -0.016$, $p = 0.894$, over 76 standards with ≥ 10 development problems). This dissociation suggests that the problem-level frequency effect reflects the intrinsic difficulty of problems that hap-

pen to target rare standards, rather than the model simply failing due to insufficient training signal for those standards. Pedagogical subtlety and data sparsity are thus partially confounded in this benchmark, and both likely contribute to the performance gaps we observe.

Error analysis. Qualitative error analysis of M3’s failure cases reveals three major error patterns. First, *incomplete prediction*: the system often identifies a primary standard but misses additional relevant ones. For example, for a problem asking students to graph a quadratic function and annotate its features, the agent predicts F-IF.B.4 (interpret key features of graphs and tables, and sketch graphs showing key features given a verbal description of the relationship) but misses F-IF.C.7a (graph linear and quadratic functions and show intercepts, maxima, and minima), despite the problem explicitly requiring graphing these elements. This pattern suggests the agent stops after identifying the most salient standard without verifying completeness. It also triangulates with the low exact match at high school (0.13) where problems frequently align to multiple standards (Table 5 in Appendix E). A targeted fix for future work could be an explicit verification prompt after finalization: “Are there additional standards that also apply? Check siblings and related standards before finalizing.”

Second, *grade-level misalignment*: the system predicts conceptually related standards at the wrong grade level. For example, a problem presenting a linear relationship $y = 3x + 6$ and asking students to notice patterns falsely received the high school standard prediction F-IF.B.4 (interpret key features of graphs and tables given a function modeling the relationship between two quantities), rather than the 8th grade standard 8.F.B.4 (construct a function to model a linear relationship from a table). Because the system lacks explicit grade-level calibration, and that standards across grades often share closely overlapping descriptions due to the spiral nature of math curricula (where later standards build upon earlier ones) (Ireland and Mouta, 2020), it fails to recognize that 8.F.B.4 emphasizes constructing linear functions from tables as an introductory concept, while F-IF.B.4 involves more sophisticated interpretation of more diverse functions and relationships. Future work could incorporate grade-level filtering or an explicit complexity classification step.

Third, *sibling confusion*: the system still con-

flates structurally adjacent but pedagogically distinct standards within the same cluster. For instance, for a problem asking which scenarios require multiplying $\frac{1}{8} \times \frac{2}{5}$, the agent predicts 5.NF.B.4 (apply and extend previous understandings of multiplication to multiply fractions) when the gold includes 5.NF.B.6 (solve real world problems involving multiplication of fractions and mixed numbers). In this case, while both standards share the same parent cluster (5.NF.B), there is a distinction between *applying* the operation versus *solving* contextualized problems requiring it. This validates the MathFish paper finding that models predict labels that are “close to ground truth, but differ in subtle ways” (Lucy et al., 2024), and also explains the weak performance on Functions (0.09) and Fractions (0.10) as domains with dense clusters of similar-sounding standards. A targeted improvement could be to augment the `get_standard_detail` tool of the ReAct agent with contrastive descriptions for common confusion pairs.

Implications for practitioner workflows. The pipeline offers two natural operating points for educators. M2’s top-5 ranked list (weak accuracy 0.688) supports a *suggest-and-verify* workflow in which a reviewer scans a short candidate set rather than searching across all 385 standards manually. M3’s variable-size output (exact match 0.313, weak accuracy 0.589, graph distance 0.93) supports a *draft-tagging* workflow in which the system proposes an initial alignment that humans then audit. Because M3’s graph distance of 0.93 means that even incorrect predictions typically land within the same cluster or an adjacent one, the expected correction effort is a local adjustment (e.g., swapping a sibling standard) rather than a full re-search of the taxonomy. From an instructional design perspective, not all error types carry equal cost: grade-level misalignment is arguably the most consequential because it mistargets developmental appropriateness, whereas sibling confusion preserves the correct instructional focus area, and incomplete prediction falls between the two since the primary standard is usually identified. These distinctions suggest that future deployment should surface confidence signals alongside predictions so that reviewers can prioritize verification effort accordingly.

Curriculum alignment as a fundamentally hard problem. These results underscore that curriculum alignment is harder than mathematical rea-

soning. Solving a problem requires producing a correct answer, but aligning it to standards requires inferring the pedagogical intent behind its construction, the specific competencies it targets, and how those relate to structurally adjacent standards in the curriculum hierarchy (Lucy et al., 2024; Sonkar et al., 2024). The spiral nature of math curricula (Ireland and Mouthaan, 2020), where later standards build upon earlier ones with overlapping descriptions, further compounds this difficulty. This explains why embedding-based retrieval and reranking is insufficient even with strong fine-tuned encoders (Xu et al., 2025) and why even our best system with agentic reasoning capabilities achieves only 31.3% exact match. Nonetheless, it is worth noting that out of the cases with identified errors, more than half of them achieve weak match, and M3’s graph distance of 0.93 already substantially improves over GPT-4’s 1.90 (Lucy et al., 2024), suggesting the system is frequently in the right neighborhood. This raises the question of whether exact match is the right criterion in the first place: curriculum standards alignment requires judging whether a problem enables students to learn the “full intent” of a standard—a nuanced judgment that Lucy et al. (2024) suggest professional curriculum reviewers may not always make consistently, though the degree of inter-annotator disagreement remains unquantified in the literature. Future work should move beyond binary exact match toward evaluation protocols that assign partial credit weighted by curriculum proximity, and measure human-human and human-AI agreement rather than accuracy against a single gold standard, better reflecting the inherently subjective nature of curriculum alignment.

6 Conclusion

We studied automated curriculum alignment on the MathFish benchmark, testing whether training-based retrieval and agentic reasoning can improve upon prompting-only baselines. Our ablation shows that contrastive retrieval (M1) and cross-encoder re-ranking (M2) provide strong candidate coverage but cannot predict exact matching curriculum standard sets on their own, both producing 0.000 exact match. This is expected given that their prediction sets are fixed at 20 and 5 candidates respectively, far exceeding the gold average of 1.47. Attaching a ReAct agent and LLM-as-a-judge critic (M3) raises exact match to 0.313,

approximately 6.5 times the GPT-4-Turbo baseline (Lucy et al., 2024). Notably, a single-pass LLM-as-a-judge critic without the ReAct loop already achieves 28.4%, reflecting that M1+M2’s role is to transform the 385-way alignment problem into a focused pruning task over a small, high-quality candidate set, in contrast to the prompt-only baseline (Lucy et al., 2024), which requires the LLM to navigate the full taxonomy unaided. The ReAct loop’s iterative tool use then adds a further +2.9 percentage points on top.

Limitations

As demonstrated, even our best system achieves only 31.3% exact match, and error analysis reveals three systematic failure modes: incomplete prediction, grade-level misalignment, and sibling standards confusion. These failures are partly structural: the spiral nature of math curricula means that standards across grades deliberately share overlapping language, making grade-level disambiguation fundamentally harder than lexical similarity would suggest. The system also has no mechanism to reason about the cognitive demand or instructional context of a problem, both of which human reviewers draw on when distinguishing, for example, a standard targeting procedural fluency from one targeting conceptual understanding within the same cluster. Additionally, the 40/30/20/10 hard negative sampling ratio in M1 is a fixed heuristic informed by prior error analysis rather than an empirically optimized hyperparameter, and we leave ablation of this ratio to future work. A related concern is that sibling standards sampled as hard negatives may occasionally be true positives for a given problem, potentially teaching the bi-encoder to over-suppress genuinely related standards and contributing mechanistically to the incomplete prediction failure mode we identify in Section 5. Quantifying and filtering contaminated negatives is an important direction for future work. Finally, the agent and critic in M3 share the same underlying model, limiting critic independence: while the asymmetric prompting—the agent explores, the critic prunes—produces empirically observable reduction in predictions (5.00 to 1.37), systematic errors rooted in the model’s parametric representation of CCSS standards, such as grade-level misalignment and sibling confusion, are unlikely to be caught by a critic sharing those same representations. Future work should explore using a different model family

or a trained discriminative verifier as the critic.

Beyond the model itself, exact match may not be the right evaluation criterion for this task. Curriculum alignment is inherently a judgment call: while Lucy et al. (2024) note qualitatively that professional reviewers may disagree, neither that work nor ours quantifies inter-annotator agreement directly, and establishing a human agreement ceiling remains important future work. Future work should also develop evaluation protocols that assign partial credit weighted by curriculum proximity rather than treating a single gold annotation as ground truth. On the modeling side, incorporating grade-level signals, contrastive standard descriptions for common confusion pairs, and explicit completeness verification steps could address the identified failure modes more directly.

Ethical Considerations

Curriculum alignment systems used in educational settings have real consequences for both students and teachers. Incorrect standard predictions can result in poorly targeted instruction or assessments, and may disproportionately impact students in under-resourced schools where automated tools are more likely to replace expert review. We advise against deploying these systems without human oversight. In practice, they should be co-developed with educators and curriculum specialists to ensure the outputs are pedagogically sound and interpretable, and they should be evaluated against the judgment of domain experts rather than relying solely on proposed metrics.

References

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Common Core State Standards Initiative. 2010. Common core state standards for mathematics. <https://www.corestandards.org/Math/>.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. [From local to global: A graph rag approach to query-focused summarization](#). *Preprint*, arXiv:2404.16130.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and

- Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.
- Jo Ireland and Melissa Mouthaan. 2020. [Perspectives on curriculum design: comparing the spiral and the network models](#).
- Amay Jain, Liu Cui, and Si Chen. 2025. [Aligning llms for the classroom with knowledge-based retrieval – a comparative rag study](#). *Preprint*, arXiv:2509.07846.
- Zhi Li, Zachary A. Pardos, and Cheng Ren. 2024. [Aligning open educational resources to new taxonomies: How ai technologies can help and in which scenarios](#). *Computers Education*, 216:105027.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. [Sparse, Dense, and Attentional Representations for Text Retrieval](#). *arXiv preprint*. ArXiv:2005.00181 [cs].
- Li Lucy, Tal August, Rose E. Wang, Luca Soldaini, Courtney Allison, and Kyle Lo. 2024. [Math-Fish: Evaluating language model math reasoning via grounding in educational curricula](#). *Preprint*, arXiv:2408.04226.
- Stephen Robertson and Hugo Zaragoza. 2009. [The Probabilistic Relevance Framework: BM25 and Beyond](#). *Found. Trends Inf. Retr.*, 3(4):333–389.
- Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Sean McGrew, and Dongwon Lee. 2021. [Classifying math knowledge components via task-adaptive pre-trained bert](#). In *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14-18, 2021, Proceedings, Part I*, page 408419, Berlin, Heidelberg, Springer-Verlag.
- Yuhong Shi, Kun Yu, Yifei Dong, and Fang Chen. 2026. [Large language models in education: a systematic review of empirical applications, benefits, and challenges](#). *Computers and Education: Artificial Intelligence*, 10:100529.
- Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard G. Baraniuk. 2024. [Pedagogical Alignment of Large Language Models](#). *arXiv preprint*. ArXiv:2402.05000 [cs].
- Student Achievement Partners. 2024. Achieve the core. <https://achievethecore.org/>.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. [Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models](#). *Preprint*, arXiv:2104.08663.
- Venkatesh V, Mukesh Mohania, and Vikram Goyal. 2021. [Tagrec: Automated tagging of questions with hierarchical learning taxonomy](#). *Preprint*, arXiv:2107.10649.
- Venkatesh Viswanathan, Mukesh Mohania, and Vikram Goyal. 2022. [Tagrec++: Hierarchical label aware attention network for question categorization](#). *Preprint*, arXiv:2208.05152.
- Qingshu Xu, Hong Jiao, Tianyi Zhou, Ming Li, Nan Zhang, Sydney Peters, and Yanbin Fu. 2025. [Automated alignment of math items to content standards in large-scale assessments using language models](#). *Preprint*, arXiv:2510.05129.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing Reasoning and Acting in Language Models](#). *arXiv preprint*. ArXiv:2210.03629 [cs].
- Ozgur Yilmazel, Niranjan Balasubramanian, Sarah Harwell, Jennifer Bailey, Anne Diekema, and Elizabeth Liddy. 2007. [Text categorization for aligning educational standards](#). page 73.
- Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. 2022. [Use all the labels: A hierarchical multi-label contrastive learning framework](#). *Preprint*, arXiv:2204.13207.
- 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](#). *arXiv preprint*. ArXiv:2306.05685 [cs].

A Prompts for ReAct Agent with LLM-as-a-Judge

A.1 ReAct agent prompt

◇ ReAct system prompt for M3 and A1

You are an expert at aligning math problems to Common Core State Standards.

Task. Given a math problem and (optionally) an initial list of candidate standards, decide which standards the problem aligns with. A problem may align to multiple standards if it addresses multiple concepts or skills. You may use tools to:

1. `get_standard_detail(standard_id)` – fetch full description and metadata for one standard.
2. `get_related_standards(standard_id, relation)` – fetch related standards (siblings in same cluster or conceptual links; relation is “siblings” or “conceptual”).
3. `search_standards(query)` – when the initial candidates look wrong, search again with a short phrase (e.g., “fractions with like denominators”, “linear equations grade 8”) to find more candidates.

Respond in this exact format at every step:

- Thought: <your reasoning>
- Action: <exactly one tool call: `tool_name(arg1, arg2)`>

When you are done and ready to give the final set of standard IDs:

- Thought: <brief reasoning summarizing why these standards are correct and complete>
- Final Answer: <comma-separated list of standard IDs, e.g., 7.NS.A.1c, 7.EE.A.1, or none>

Rules.

- Only output standard IDs that appear in tool results (initial candidate list, `get_standard_detail`, `get_related_standards`, or `search_standards`). Do not invent new IDs.
- A problem aligns with a standard only if students could reasonably learn the full intent of that standard from the problem, not just see a related idea in passing.
- Include all standards that the problem truly aligns with; if a problem teaches multiple concepts, list all relevant IDs.
- If you are not confident that any standard clearly meets this bar, output Final Answer: none.
- Use exact CCSS IDs (e.g., 7.NS.A.1c, A-REI.D.11).

A.2 Critic prompt

◇ Critic prompt (LLM-as-a-judge) for pruning predictions for M3 and A1

You are an expert at aligning math problems to Common Core State Standards.

You will be given:

1. A math problem.
2. A list of *candidate* standards (ID + description) that another agent thinks might apply.

Your job is **only** to prune this candidate list:

- You may only select from the given IDs. Do *not* invent or add new standard IDs.
- A problem aligns with a standard only if it can enable students to learn the full intent of that standard’s description, not just mention a related idea in passing.
- Choose the smallest subset of IDs that directly match what the problem is assessing or teaching.
- If two standards are clearly redundant (same idea, different grade bands), keep the best match and drop the others.
- If you are not confident that any candidate clearly meets this bar, prefer none over guessing. If none of the candidates are good matches, you may output none.

Respond in this format **only**:

- Thought: <your reasoning>
- Final Answer: <comma-separated list of chosen IDs, or ‘none’>

B Preliminary Pipeline Variants (100-Problem Development Subset)

All variants A–F use `gemini-2.0-flash-001` with the same ReAct agent and critic, and pass top-25 retrieved candidates to the agent; only the retrieval stage (and optionally a verifier) differs. Each was evaluated on a fixed 100-problem subset of the Addressing/Alignment-filtered development set (all problems have non-empty gold labels; $N = 100$ for all rows).

- A No retrieval (no RAG).
- B BM25 only.
- C Hybrid retrieval (BM25 + dense).
- D Hybrid retrieval + ATC curriculum graph reranking (cluster and domain proximity).

- E** Graph-first retrieval: hybrid seeds ($k_{\text{seed}} = 3$) expanded via the ATC curriculum graph.
- F** Same as E, with an additional per-standard verifier (Yes/No) before the critic.

Model ablation. Variant D* replicates pipeline D but substitutes gemini-2.5-flash-lite for gemini-2.0-flash-001 to ablate the effect of model choice; it is substantially slower and predicts more standards per problem on average, but modestly improves weak accuracy.

Table 2 reports results; **the selected pipeline is Variant D.**

Table 2: Preliminary results on the fixed 100-problem development subset ($N = 100$). Bold indicates best per metric. Graph Distance (GraphDist) and Sibling Confusion Rate (SiblingConf) are graph-based alignment quality metrics where lower is better.

Variant	Exact	Weak Acc	Micro F1	Macro F1	R@5	R@20	GraphDist	SiblingConf
A	0.26	0.48	0.38	0.39	—	—	1.13	0.10
B	0.23	0.48	0.36	0.37	0.25	0.43	1.23	0.15
C	0.28	0.59	0.42	0.46	0.37	0.69	0.95	0.15
D	0.30	0.64	0.46	0.50	0.39	0.70	1.13	0.15
E	0.24	0.57	0.41	0.43	0.28	0.48	1.04	0.14
F	0.30	0.54	0.41	0.44	0.29	0.46	1.11	0.17
D*	0.17	0.70	0.35	0.43	0.40	0.70	1.49	0.08

Variants A–F use gemini-2.0-flash-001; D substitutes gemini-2.5-flash-lite on the selected pipeline (D) as an ablation on model choice.

C Example from MathFish Benchmark (Lucy et al., 2024)

◇ Example

Problem. Consider a circle with center O and let P be a point on the circle. Suppose L is a tangent line to the circle at P , that is, L meets the circle only at P . Show that $\overline{OP}$ is perpendicular to L .

Output Standards:

- **G-C.A.2** Identify and describe relationships among inscribed angles, radii, and chords (e.g., tangent line perpendicular to radius).
- **G-CO.C.9** Prove theorems about lines and angles (e.g., tangent perpendicular to radius).
- **G-CO.A** Know precise definitions of angle, circle, line, parallel line, and line segment.

D Label Frequency Distribution

Table 3 reports the distribution of training examples per standard in the Addressing/Alignment-filtered training set. Of the 385 standards in the full CCSS taxonomy, 584 appear at least once across the 5,956 training problems.³ Distribution is right-skewed: the median standard has 11 training examples, but 20.7% have fewer than 5 and 44.0% have fewer than 10, while the most frequent standard appears in 139 problems.

³The count exceeds 385 because multi-label problems increment each appearing standard, and sub-standard labels introduced via inheritance (Appendix A.2 of Lucy et al. (2024)) expand the effective label set.

Table 3: Distribution of training examples per standard (n_s) across standards with at least one training example in the Addressing/Alignment-filtered training set (5,956 problems).

Training examples (n_s)	# standards	% of standards
< 2	32	5.5%
2–4	89	15.2%
5–9	136	23.3%
10–19	174	29.8%
20–50	114	19.5%
> 50	39	6.7%
Total	584	100%

Summary statistics: min = 1, median = 11, mean = 15.0, max = 139.

Table 4 reports M3 exact match stratified by training frequency of the rarest gold standard per problem ($m = \min_{s \in G} n_s$) and the two correlation analyses described.

Table 4: M3 exact match by training frequency quartile. Problems are binned by m , the training count of the rarest gold standard in their gold set. Quartiles are equal-frequency (approximately 485–486 problems each).

Quartile	n	EM	min m	max m
Q1 (lowest m)	486	0.148	0	10
Q2	485	0.332	10	18
Q3	485	0.324	19	31
Q4 (highest m)	486	0.449	31	139
Overall	1,942	0.313	–	–

Spearman ρ between $\log(1 + m_i)$ and exact match (per problem, $n = 1,942$): $\rho = 0.234$, $p = 1.31 \times 10^{-25}$. Spearman ρ between $\log(1 + n_s)$ and conditional exact-match rate (per standard, $n = 76$ standards with ≥ 10 dev problems): $\rho = -0.016$, $p = 0.894$.

E M3 Performance by Grade and Domain

Tables 5 and 6 report exact match disaggregated by grade level and domain for M3 (BiEncoder + Cross-Encoder Rerank + ReAct) on the full Addressing/Alignment-filtered development set ($N = 1942$).

Table 5: Exact match by grade level on the full Addressing/Alignment-filtered development set. N counts problem–grade pairs.

	K	1	2	3	4	5	6	7	8	HS
N	166	221	211	317	326	340	396	375	387	1027
M3: BiEncoder + Cross + ReAct	0.27	0.29	0.30	0.26	0.22	0.12	0.19	0.21	0.22	0.13

Table 6: Exact match by domain on the full Addressing/Alignment-filtered development set. N counts problem–domain pairs.

	C&C	Func	Geom	M&D	NOBT	NOF	NSQ	O&A	R&P	S&P
N	72	407	646	230	465	254	214	914	217	347
M3: BiEncoder + Cross + ReAct	0.19	0.09	0.30	0.23	0.32	0.10	0.19	0.18	0.16	0.24

*Note: C&C = Counting & Cardinality, Func = Functions, Geom = Geometry, M&D = Measurement & Data, NOBT = Number & Operations in Base Ten, NOF = Number & Operations – Fractions, NSQ = Number Systems & Quantity, O&A = Operations & Algebra, R&P = Ratios & Proportional Relationships, S&P = Statistics & Probability.

Through the Sentence Lens: Explainable Essay Scoring through Fine-Grained Predictions

Daniel Mora^{1,2}, Stefan Keller³, Andrea Horbach^{1,2}

¹Leibniz Institute for Science and Mathematics Education, Kiel, Germany

²Kiel University, Kiel, Germany,

³Zurich University of Teacher Education, Zurich, Switzerland

Correspondence: mora@leibniz-ipn.de

Abstract

Beyond performance, model transparency is a crucial factor in Automated Essay Scoring, yet current systems often lack explainability, limiting their pedagogical value and users' trust. Existing explainability methods, such as gradient-based attribution or feature-importance approaches, either produce counterintuitive explanations or are too complex for classroom use. To address this limitation, we make use of fine-grained prediction at the sentence level as a way to enhance explainability. We propose ablation strategies to derive sentence-level pseudo scores from essay-level gold scores and use them to train sentence-level models. We evaluate their performance against essay-level baselines on two datasets (ASAP and MEWS), and compare their sentence-level output to a human baseline. Results indicate a trade-off between essay-level performance and sentence-level granularity. For the *language quality* trait, most sentence-level models achieve performance comparable to the essay-level baseline, whereas for *content*, the approach yields more positive results on prompts with shorter student texts.

1 Introduction

Automated Essay Scoring (AES) systems are tools designed to evaluate free-text written responses by assigning a score typically using machine learning algorithms. Despite their significant benefits for teachers, such as reducing the workload and making it possible to provide timely feedback to students, their implementation in real-life educational scenarios remains limited due to the lack of explainability, that is, the model's reasoning cannot be made understandable to humans (Li et al., 2022). Whether used for formative feedback or in high-stakes exams, improving such transparency for end users is essential to enhance the overall usability of AES systems (Hall et al., 2024).

Most approaches to improving model explainability have focussed on developing inherently interpretable models or applying post-hoc local explanation methods. However, these techniques often lack pedagogical value in educational contexts: gradient-based attribution methods can be counterintuitive (Parekh et al., 2020), and feature-importance approaches (Kumar and Boulanger, 2020; Urrutia et al., 2025) frequently suffer from construct validity issues (Beigman Klebanov and Madnani, 2020). Rubric-based or fine-grained predictions could offer more intuitive explanations for teachers and students, but their development is constrained by the scarcity of annotated datasets (Ramesh and Sanampudi, 2022), which limits the training of machine learning models.

To address this gap in explainability, we build on the idea of fine-grained prediction by focusing on scoring each trait at the sentence level, rather than assigning only an overall holistic score. Because there is a lack of sentence-level annotations, we propose a method for deriving sentence-level pseudo-scores from essay-level annotations using an ablation strategy. Specifically, we use a model trained at the essay-level to predict the relevance of a sentence by comparing the difference between the prediction on the full essay and the prediction when that particular sentence is missing. This difference in prediction indicates whether the sentence contributes positively or negatively to the overall score.

As a next step, we train sentence-level models using the pseudo scores we had derived. Table 1 illustrates an example output showing sentence-level predictions that can be useful for learners' writing, as receiving feedback on individual sentences helps them focus their efforts when revising their texts. Finally, we evaluate the proposed approach by comparing the performance of sentence-level models against essay-level baselines and by evaluating the automated sentence-level scores against

human annotations¹.

This research thus makes two contributions:

- **Model Performance and Ablation Impact:**

We compare the performance of different AES models architectures trained at the essay level using annotated gold scores with models trained at the sentence level using pseudo scores generated through ablation strategies on ASAP and MEWS datasets.

- **Alignment with Human Judgements:** We further examine the extent to which the sentence rankings predicted by sentence-level models align with human evaluations of sentence quality on a subset of the MEWS dataset.

Text	Score
Older children, like with the age of five or six or elder, it is of course right and possible to watch television with a time, which is given by the parents, so that the children can watch a serie for one hour and then it's finished for this day.	4.1
So all in all, I agree at some parts with the statement, but not at all.	1.4
The main thing, that is important is the bad influence by television for too young children, so it could exists problems in future, which are developed by the television.	0.9

Table 1: Sentence-level predictions for the last three sentences of a random essay in the MEWS dataset (Keller et al., 2020). Score range is 0–5.

2 Related Work

Research on transparency in AES has taken several directions, focusing on model interpretability, model explanations, trait scoring, or fine-grained predictions.

2.1 Model Interpretability

Model interpretability refers to explaining the intrinsic properties of the model that make it understandable to humans (Doshi-Velez and Kim, 2017). As the computational complexity of a model increases, however, its inherent interpretability decreases (Barceló et al., 2020; Bexte et al., 2024). For example, in a linear regression model, the prediction is expressed as a weighted sum of features, where each coefficient reflects the relevance of a feature (Molnar, 2025).

¹Source code available at <https://github.com/melanchthon19/sentence-level-aes>

In this context, Urrutia et al. (2025), Kumar et al. (2019), and Eltanbouly et al. (2025) use linear regressor and Random Forest models due to their higher degree of intrinsic interpretability and provide the most relevant features as feedback to the learner. Although complex models tend to outperform simpler ones, this is not always the case (Li and Ng, 2024), and simpler models should be preferred if interpretability is a main concern (Rudin, 2019).

The use of generative LLMs in the AES context has been widely explored, yielding mixed results (Mizumoto and Eguchi, 2023; Lee et al., 2024; Mansour et al., 2024; Yoshida, 2025). Even though generative models are able to produce explanations using natural language when instructed to provide insights into their decision-making process, these explanations could not reflect the model’s underlying reasoning, and can be yet misleading (Turpin et al., 2023). Methods such as chain-of-thought prompting and self-explanations have been proposed to improve explainability, but they remain limited, as they do not provide direct access to the internal decision process of the model (Lanham et al., 2023; Dehghanighobadi et al., 2025).

2.2 Model Explanations

If a model is too complex, an interpretation algorithm (Li et al., 2022) can be applied to reveal how the model makes decisions, usually in the form of a post-hoc local explanation (Madsen et al., 2022). Common approaches for explaining neural networks are attention-based (Dong et al., 2017; Yang et al., 2020), gradient-based (Alikaniotis et al., 2016), and feature-based attribution methods (Kumar and Boulanger, 2020).

Although post-hoc local explanations can be useful, their pedagogical value remains limited. For example, gradient-based attributions may highlight factors that do not align with human intuitions, such as different sub-tokens of the same word contributing to opposite predictions (Chefer et al., 2021), or the order and even the presence of certain words being irrelevant to the model (Parekh et al., 2020). Similarly, in feature-based attribution methods, the identified feature may be only an indirect proxy that is not explicitly assessed, such as essay length (Jeon and Strube, 2021), or it may be so heavily engineered that the model loses decomposability (Lipton, 2018).

2.3 Trait Scoring

Trait or rubric-based scoring disaggregates a holistic score into specific traits defined in a rubric. Mathias and Bhattacharyya (2018) enrich the ASAP dataset with trait-level annotations, and Yoo et al. (2025) extend this line of work by exploring augmentation strategies. Other studies focus on modelling approaches: Mizumoto et al. (2019) propose a BiLSTM model combined with supervised attention for justification identification; Kumar and Boulanger (2021) examine the relationship between groups of linguistic features and specific traits, as well as their contribution to holistic scoring; Lohmann et al. (2024) compare models and input features on trait scoring; and Eltanbouly et al. (2025) leverage generative LLMs to extract rubric-based features. While trait scoring provides more explanations than a single holistic score, the assigned scores remain at the essay level, with no finer-grained evaluation.

2.4 Sentence-Level Predictions

In addition to trait-level scoring, explainability can also be enhanced through fine-grained prediction, where models score sentences instead of providing just one overall score. Because manual annotations at the sentence level are scarce and expensive, most supervised approaches must circumvent the lack of training data.

Andersen et al. (2013) frame the task as discriminative ranking and compare two approaches. The first uses an essay-level model to rank sentences, which produces weak results. The second derives pseudo sentence scores by dividing holistic essay scores by the number of manually annotated errors per sentence, which yields stronger sentence-ranking performance. When aggregating back to the essay level, however, the sentence-level model underperforms compared to the model trained directly on holistic essay scores. No direct evaluation of the sentence rankings is reported.

Woods et al. (2017) propose an ablation strategy that avoids fine-grained sentence-level annotations. Using an essay-level model, they estimate sentence contributions by omitting sentences and measuring changes in predicted scores. To account for length bias, essay length is included as a fixed feature, and score changes are evaluated against a background distribution of unannotated essays. Results show a slight advantage over a baseline in identifying trait-relevant sentences, though no direct comparison is

made with the original essay-level model.

Hossain and Mustafa (2023) explore the use of an adjacent sentence attention mechanism designed to capture contextual dependencies between neighbouring (preceding and following) sentences. Their model underperforms compared to pretrained baselines, mainly due to the scarcity of training data.

These approaches have some limitations: they either do not directly evaluate the performance of the sentence-level model, lack comparison with essay-level models trained on manual annotations, or are constrained by small datasets. This study addresses these gaps by training sentence-level models using pseudo scores, evaluating their performance against essay-level baselines, and validating sentence-level outputs through a human annotation study.

3 Datasets

We use the ASAP² dataset and its trait-enriched extension ASAP++ (Mathias and Bhattacharyya, 2018), as well as the MEWS (Keller et al., 2020, 2024) dataset. Both corpora cover a wide range of argumentative and narrative writing tasks, and represent diverse student populations. ASAP consists of essays written by native English students from grades 7 to 10 for 8 different prompts, whereas MEWS represents English L2 learners from German and Swiss upper secondary schools with German as L1 for two prompts (TE and AD). In our experiments we also considered their combination (TEAD). Further details of the datasets are provided in the original articles.

For consistency across datasets, we select prompts that include comparable trait-level annotations. We also exclude document-level traits such as *holistic* and *organisation* given that evaluating their quality goes beyond the sentence unit.

The final set of traits used in our experiments include *content* and *language quality*. According to the evaluation rubrics, these traits include effective language command, accurate syntactic and idiomatic usage, sufficient and engaging details, relevant explanations, and well-supported ideas. Table 2 summarises the score distribution for the two traits across prompts, which follow an approximately bell-shaped distribution.

²<https://www.kaggle.com/datasets/lburleigh/asap-2-0>

Prompt	Count	Lang. Qual.	Content
MEWS			
AD	1169	2.79 ± 1.10	3.35 ± 0.96
TE	1099	3.12 ± 1.18	3.68 ± 0.84
ASAP			
1	1783	–	3.85 ± 0.99
2	1801	–	3.22 ± 1.17
3	1726	1.47 ± 0.85	1.43 ± 0.84
4	1770	1.06 ± 0.88	1.11 ± 0.97
5	1805	2.24 ± 0.94	1.88 ± 1.00
6	1800	2.05 ± 0.92	1.85 ± 1.11
7	1569	–	1.84 ± 0.80
8	723	–	3.80 ± 0.63

Table 2: Label distribution of *language quality* and *content* (mean and SD) per prompt for MEWS and ASAP.

4 Experimental Study: Sentence-Level Predictions

In order to evaluate if fine-grained essay scoring improves explainability while preserving the scoring performance, we (i) train essay-level models on the manually annotated scores, (ii) generate sentence-level pseudo scores through ablation strategies, (iii) train sentence-level models using these pseudo scores, and (iv) evaluate the aggregated essay-level predictions against the human-annotated gold standard using Quadratic Weighted Kappa (QWK) (Cohen, 1968). Figure 1 depicts the overall process.

4.1 Models

We train essay-level models representing the main AES paradigms (Xu et al., 2024): linguistic feature-based, transformer-based, and hybrid models. The corresponding architectures are shown in Figure 2 and training hyperparameters in Appendix A.1. Sentence-level models use the same architectures and hyperparameters as their corresponding essay-level baselines.

Linear Regressor (LR): This model is trained on 220 handcrafted linguistic features extracted using the tool described in Lohmann et al. (2024). The features include lexical, syntactic, cohesion-related, error-based, frequency-based, and length-based indicators. The features are scaled and multicollinear features are removed before being fed into the model.

DistilBERT (LLM): A DistilBERT³ model (Sanh et al., 2019) with pretrained weights is used to encode the input essay. The essay representation is obtained by applying max pooling to the final hidden layer, and it is then passed to a custom regression head composed of two linear layers, which

is fine-tuned on top of the encoder.

Hybrid Model (Hybrid): This model integrates the first two approaches by combining the max pooled representation of the final hidden layer from the DistilBERT encoder with a feedforward network that processes the 220 linguistic features used in the linear regression model. The two representations are concatenated and passed through a shared regression head composed of two linear layers.

4.2 Pseudo Scores via Ablation

The main intuition is that the quality of an essay is not equal across all sentences, and that some sentences have a stronger influence on a positive or negative score. Removing a sentence should cause the model to change its prediction according to the relevance of the sentence.

Using the model trained at the essay level, we estimate the magnitude and direction of this relevance by computing the difference in predictions on ablated (missing one or more sentences) and full essays. If the prediction of the ablated essay is higher compared to that of the full essay, it means that the missing fragment (one or more sentences) was lowering the score. Therefore, this fragment should be assigned a lower pseudo score compared to the score assigned at the essay level.

Once pseudo scores are generated, they are normalised to match the target scoring range. Normalisation proceeds in three steps. Z-score normalisation centres the values at zero and scales them to have a unit standard deviation. Tanh non-linear transformation then compresses outliers beyond one standard deviation, and, finally, min-max scaling maps the transformed values to the target scoring range. Figure 6 in Appendix A.2 illustrates the resulting distribution of pseudo scores after normalisation.

4.2.1 Sentence Ablation

To generate sentence-level pseudo scores, each essay is first segmented into individual sentences using spaCy (Honnibal et al., 2020). Then, the ablation strategy follows the leave-one-out method by removing one or more sentences at a time, predicting a score on the ablated essay, and computing a pseudo score for that particular fragment using the following formula:

$$\hat{y} = y - \alpha \cdot (\hat{y}_{\text{loo}} - \hat{y}_{\text{essay}} - b), \quad (1)$$

where $\hat{y}$ is the pseudo score, y is the true essay-level score, $\hat{y}_{\text{loo}}$ is the prediction without the target

³distilbert-base-uncased

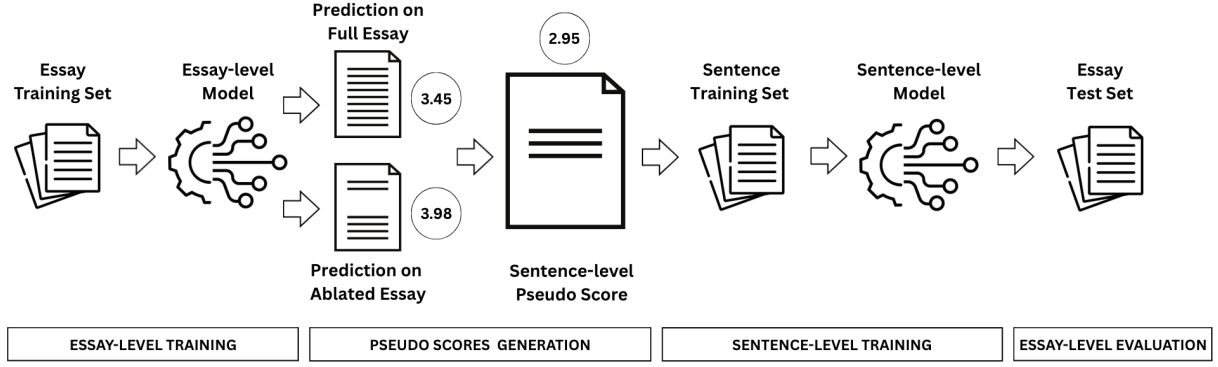


Figure 1: Pipeline for training sentence-level models using pseudo sentence scores derived from essay-level annotations via ablation. The pseudo scores generation is described in Section 4.2.

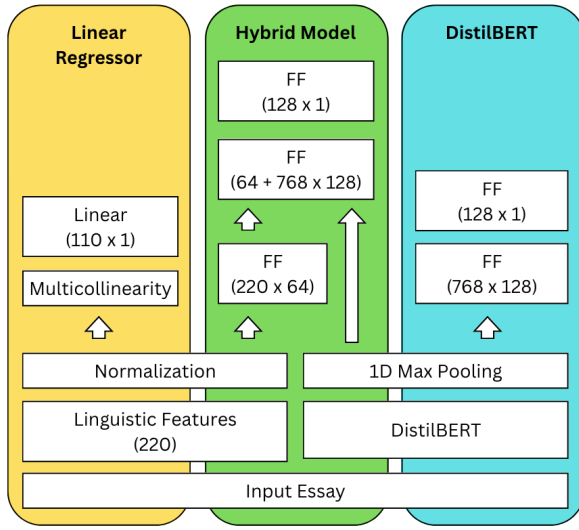


Figure 2: Model architectures used for training both essay- and sentence-level models.

fragment (leave-one-out), $\hat{y}_{\text{essay}}$ is the prediction on the full essay, and b is a bias term estimated as the mean difference between $\hat{y}_{\text{loo}}$ and $\hat{y}_{\text{essay}}$ across the entire dataset, which accounts for the model’s tendency to prefer longer inputs with more context. A scaling factor α amplifies the contribution difference, given that the model is generally stable on ablated essays, and its predictions have a low variance within the same essay. For the experiments, we varied α across the values 1, 10, 20, and 30.

For multi-sentence ablation, a sliding window n is used for leaving out several contiguous sentences when making a prediction. The contribution of a single sentence is then computed as the average contribution across all windows in which the sentence appears. For the experiments, the window size parameters n was varied between 3 and 5.

4.3 Models’ Performance Results

Figure 3 depicts the performance of the best sentence-level models compared to their essay-level baseline across traits and prompts. The primary benefit of this comparison is to highlight the setups in which the sentence-level model trained on pseudo-scores matches the performance of the essay-level model trained on annotated gold scores while offering greater granularity.

4.3.1 Essay-Level Results

Across the three models, the ASAP dataset yields higher and more stable results (mean QWK 0.61 ± 0.08) than MEWS (0.50 ± 0.18). Across both datasets, LR performs best (average QWK 0.63), achieving the best score in 67% of cases, followed by Hybrid (0.59; best in 28% of cases) and LLM (0.50). The embeddings alone used by the LLM are not sufficient to model both *content* and *language quality* across the two datasets, compared to the other models that included linguistic features.

Comparing LR and Hybrid models, concatenating embeddings with linguistic features did not consistently improve performance, with gains observed only for three prompts. Essay baseline results suggest that using simpler models preserves performance while retaining intrinsic interpretability, as also demonstrated in Li and Ng (2024).

4.3.2 Sentence-Level Results

Language Quality: Overall, sentence-level models show similar performance compared to their essay-level baselines. The LLM model has higher QWK variance compared to LR and Hybrid models. The LLM obtains similar results on two prompts, decreases by -0.21 on average on four prompts, and increases by +0.17 on one prompt. The LR and Hybrid model achieve comparable performance on

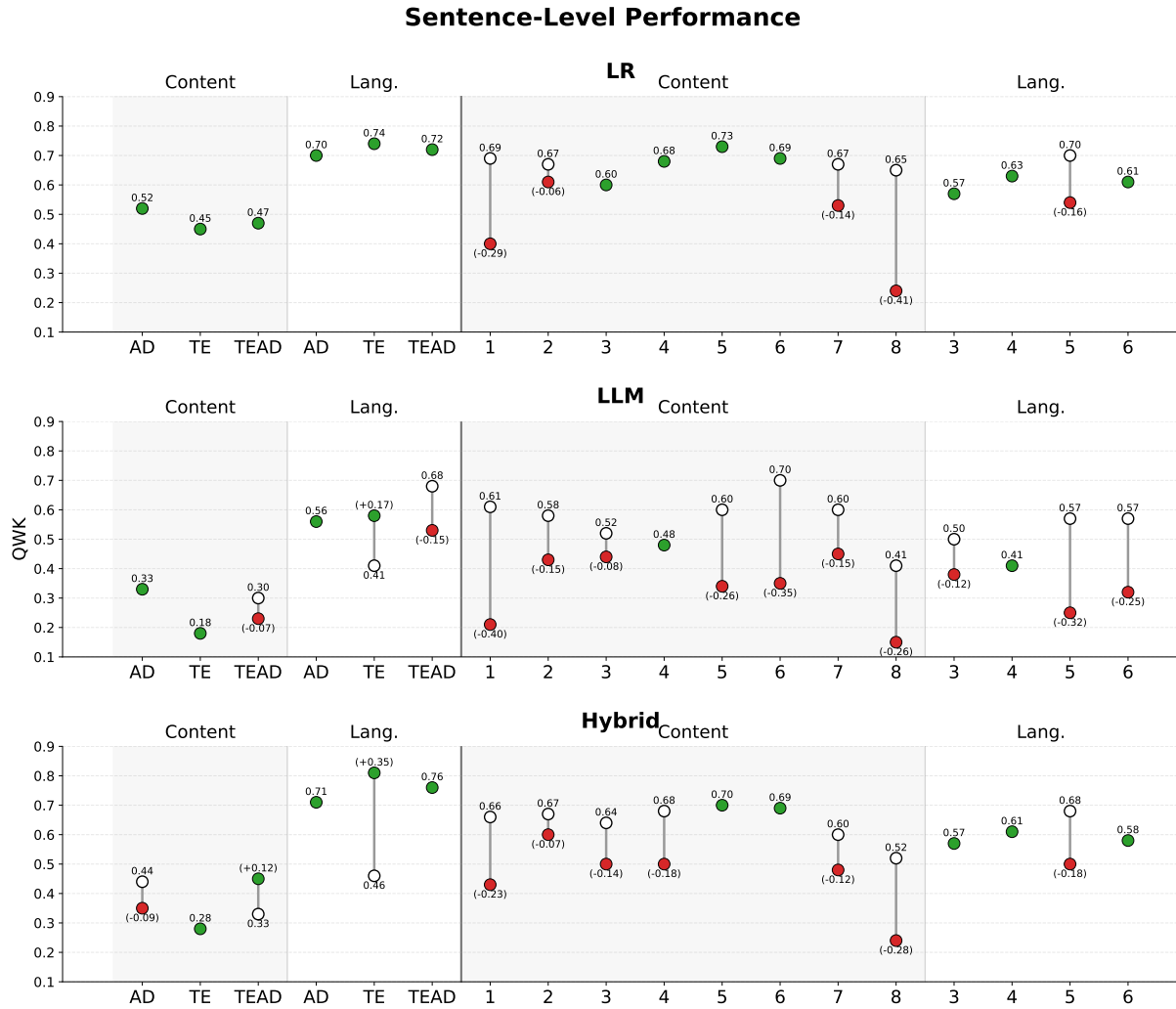


Figure 3: Performance (QWK) of the essay-level baseline (white dot) and the best sentence-level model on the ASAP and MEWS datasets per prompt and trait. All differences are statistically significant based on 95% bootstrap confidence intervals. Full results are in Appendix B.

all prompts, except on prompt 5 (ASAP), in which there is a decline of -0.17 points on average. The highest increase of +0.35 is obtained by the Hybrid model for the TE prompt (MEWS).

Figure 4 presents the confusion matrices for the essay- and sentence-level hybrid models trained on the TE prompt. The essay-level model predicts scores clustered around the middle of the scale, with a narrow range (1.5–4.0), often assigning a score of 3.0 to essays whose true scores span 1.0–5.5. In contrast, the sentence-level model, benefitting from more distributed pseudo scores, covers a wider range (1.0–5.5), and its predictions of 3.0 align more closely with true scores between 2.0 and 4.0.

Content: Results vary depending on the dataset and prompt. For the MEWS dataset, models show

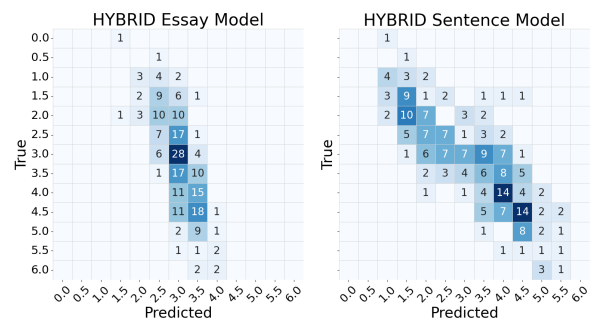


Figure 4: Confusion matrices of the Hybrid essay- and best sentence-level models trained on the TE prompt for the *language quality* trait.

similar performance compared to their essay-level baselines. Exceptions include the LLM model on the TEAD prompt and the Hybrid model on the AD prompt, both showing a decrease of less than

one point. The Hybrid model, on the contrary, shows improvement of +0.12 when both prompts are combined (TEAD).

For the ASAP dataset, models obtained a lower performance on prompts 1, 2, 7, and 8 for all three models. The LR model, however, maintains comparable performance compared to its essay-level baseline on prompts 3 to 6.

This difference in performance for the *content* trait seems to be primarily related to variation in essay length across prompts. Prompts 3 to 6 that have similar results between essay- and sentence-level models include both source-dependent and independent tasks, but the student essays for these prompts are shorter on average than those for prompts 1, 2, 7, and 8.

Ablation Strategy: A comparison of single versus multi-ablation strategies reveals that the best performances across traits and prompt setups are achieved in 69% of cases with single ablation and in 82% with an alpha value of one. The LLM model is the only one to improve further with higher alpha values, likely because its ablated predictions remain nearly constant across ablated essays, with a variance closer to 0.1. Therefore, amplifying these differences enhances the model’s ability to discriminate between sentences. For the LR and Hybrid models that use linguistic features, removing one or more sentences causes a substantial shift in the input features. This shift is sufficient to determine both the relevance and polarity of a given sentence during the ablation process, which explains why an alpha value of one was the most effective setting for these two models. The window size of 5 was beneficial to some extent (19% of cases) for all three models, with higher improvements for the *content* trait.

Sentence-level results indicate that the proposed approach can provide more detailed and informative feedback without sacrificing performance for some of the prompts and traits. This granularity allows for clearer differentiation between essays of varying quality and offers more actionable insights for teaching and learning.

5 Annotation Study: Sentence-Level Language Quality Scores

In addition to model performance comparisons, we also conduct a human annotation study. The goal is to assess whether the proposed methods agree with human judgement on identifying sentences

that contribute most positively or most negatively to the overall essay quality, focusing on *language quality* as the main criterion as it is more easily identifiable at the sentence level.

5.1 Annotation Task

To this end, five expert evaluators independently annotated a stratified random sample of 100 essays drawn from the MEWS test set. The sample was filtered according to essay length in terms of number of sentences (10–15) and character count (1,000–2,000). We manually screened the essays and excluded one partially written in a language other than English. The final sample includes 14 low-, 70 mid- and 15 high-scoring essays, evenly distributed across prompts (54 from TE and 45 from AD), resulting in 99 essays in total.

The annotators selected three sentences with the most positive impact and three with the most negative impact on the overall essay based on language command, syntactic variety, and idiomatic usage. These core aspects define the *language quality* trait as specified in the evaluation rubric.

5.2 Annotation Results

Each annotation is mapped to a numerical value (positive = 1, negative = -1, neutral = 0) and summed up across annotators. A given sentence can receive an overall score between -5 and +5, where extreme values indicate complete agreement and zero represents neutral annotations or multiple annotations balanced out. Table 3 shows an example of annotated sentences.

Text	+	-	=
That is the reason why I think, it’s impossible being nasty, if you are a friendly person normally.	0	4	-4
Another reason, why I think it’s better to have a good relation between students and teachers, is that students often learn about the behaviour of their teachers.	2	1	+1
So if a teacher wants them to learn about a subject as best without any kind of happiness or confidence, the students couldn’t learn about patience and any kind of stuff which they have to know in their future, like being comfortable and friendly.	1	4	-3

Table 3: Human annotations on three consecutive sentences from a randomly sampled essay regarding their positive (+) or negative (-) contribution to the overall essay considering *language quality* as the criterion.

The distribution of aggregated scores is shown in Figure 5: almost 20% of sentences are consid-

ered neutral (10% without any annotations and another 10% balanced out after aggregating conflicting contributions), there is higher agreement among positive sentences (score of 4 or 5), and a greater proportion of negative sentences with lower agreement (score of -1 or -2). Roughly 15% of sentences show high agreement, reaching an aggregated score greater than 4 in both the positive and negative directions. This pattern is also reflected in inter-annotator agreement measures: Krippendorff’s alpha score is 0.17 and pairwise inter-annotator Cohen’s kappa ranges from 0.14 to 0.22.

5.3 Validation of the Sentence Level: Human–Model Correlation

We evaluate the alignment between human annotations and model outputs by computing the Spearman correlation between the aggregated expert scores and the model outputs, including both the generated pseudo scores per model and sentence-level predictions. Table 4 reports the average correlation across essays and the corresponding standard deviation for both model outputs.

There is a modest Spearman correlation between the LR, LLM, and Hybrid models trained on both prompts (TEAD), based on their pseudo scores and predictions, compared with the aggregated annotation scores. The LLM model achieves the highest averaged correlation (0.29), followed by the Hybrid model (0.17), whereas the LR model shows a negative correlation (−0.15). Although the correlation is weak overall, all models nevertheless capture a meaningful underlying pattern compared to a random baseline.

The positive correlations obtained by the LLM model suggest that its sentence-level outputs are more aligned with human annotations, even though averaging sentence scores leads to lower QWKs as seen in Figure 3. This indicates that this model is better able to capture the lexical and syntactic patterns that annotators consider when judging the *language quality* of a sentence, which is likely due to the use of word embeddings. The Hybrid model also shows a meaningful correlation, though weaker, suggesting that the inclusion of linguistic features do not substantially enhance the model’s ability to identify the most positive and most negative sentences, but instead add noise.

Although the best sentence-level LR model has a similar performance compared to its essay-level baseline on 13 out of the 18 different prompts, as

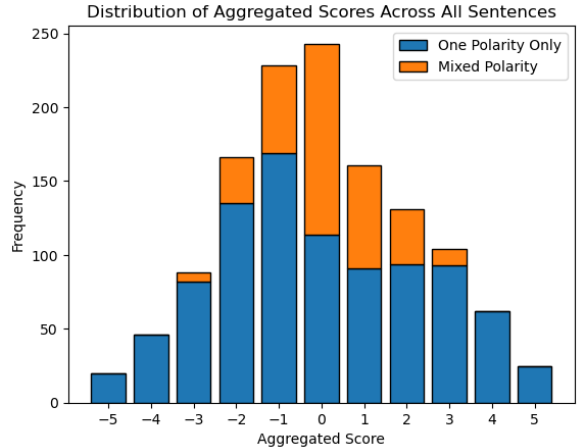


Figure 5: Distribution of aggregated sentence-level human annotations. ± 5 indicates complete agreement among all five annotators on sentence polarity.

shown in Figure 3, its sentence-level predictions diverge from human annotations with a negative correlation. This indicates that, even though aggregated sentence-level predictions correlate with overall essay scores, the individual predictions do not reflect human judgements of which sentences contribute positively or negatively. This difference can be explained by the fact that linguistic features at the sentence level are less informative, as many rely on frequencies, ratios, and distributions that require broader context.

Model	Predictions	Pseudo Scores
LLM	0.30 (0.26)	0.11 (0.25)
Hybrid	0.19 (0.28)	0.14 (0.29)
LR	-0.15 (0.28)	0.16 (0.30)

Table 4: Spearman correlations (mean and SD) between aggregated human annotations at the sentence level and models’ output (predictions and pseudo scores).

6 Conclusion

In this work, we explore fine-grained prediction as a means of enhancing the explainability of AES models. To address the lack of manual sentence-level annotations, we propose ablation strategies to generate pseudo scores for training sentence-level models and we evaluate their effectiveness against essay-level baselines using ASAP and MEWS datasets. We further examine how the outputs of these models correlate with human sentence-level annotations of the *language quality* trait on a subset of the MEWS dataset.

Overall, there is a trade-off between essay-level performance and sentence-level granularity, though it varies depending on the specific trait and ablation setup. The proposed approach is more effective for the *language quality* trait than for *content*, and for *content*, it performs better on shorter texts. Both model outputs and human judgements show relatively weak correlations, highlighting the subjective nature of the task. Despite this, averaging predictions across sentences yields strong essay-level results even when the underlying sentence-level predictions do not closely align with human reasoning. This points to the need for alternative aggregation strategies, such as assigning different weights to sentences rather than treating them all equally when averaging.

Our experiments also show that simpler models, such as a linear regressor, are able to produce competitive results at a lower computational cost and with a higher degree of intrinsic interpretability. This highlights that the trade-off between complexity and performance does not always hold, and that it might also be preferred in educational settings where transparency is crucial. Applying further explainability methods to sentence-level models would offer a clearer understanding of their internal reasoning and enhance model transparency.

Limitations

This study is limited to traits that are conveyed within the span of a sentence, such as *content* and *language quality*. The ASAP and MEWS datasets consist of English texts, and results may differ for other languages. Different encoders for the LLM and Hybrid model might have an impact on performance.

7 Ethical Considerations

All data annotation procedures were conducted by qualified annotators who were compensated in accordance with legal standards. The models were trained using data representatives of the population from which the data were obtained. Consequently, any application of these models to new or distinct populations should be undertaken with appropriate measures to ensure fairness, validity, and the minimisation of potential biases.

8 Bibliographical References

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, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. [Developing and testing a self-assessment and tutoring system](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 32–41, Atlanta, Georgia. Association for Computational Linguistics.
- Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. 2020. [Model interpretability through the lens of computational complexity](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 15487–15498. Curran Associates, Inc.
- Beata Beigman Klebanov and Nitin Madnani. 2020. [Automated evaluation of writing – 50 years and counting](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7796–7810, Online. Association for Computational Linguistics.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2024. [Strengths and weaknesses of automated scoring of free-text student answers](#). *Informatik Spektrum*, 47(3):78–86.
- Hila Chefer, Shir Gur, and Lior Wolf. 2021. [Transformer interpretability beyond attention visualization](#). In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791.
- J Cohen. 1968. [Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit](#). *Psychological bulletin*, 70(4):213–220.
- Zahra Dehghanighobadi, Asja Fischer, and Muhammad Bilal Zafar. 2025. [Can LLMs Explain Themselves Counterfactually?](#) In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7787–7815, Suzhou, China. Association for Computational Linguistics.
- Fei Dong, Yue Zhang, and Jie Yang. 2017. [Attention-based recurrent convolutional neural network for automatic essay scoring](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.
- Finale Doshi-Velez and Been Kim. 2017. [Towards a rigorous science of interpretable machine learning](#). *Preprint*, arXiv:1702.08608.

- Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. [TRATES: Trait-specific rubric-assisted cross-prompt essay scoring](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. Association for Computational Linguistics.
- Erin Hall, Mohammed Seyam, and Daniel Dunlap. 2024. Exploring explainability and transparency in automated essay scoring systems: A user-centered evaluation. In *Learning and Collaboration Technologies*, pages 266–282, Cham. Springer Nature Switzerland.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spacy: Industrial-strength natural language processing in python](#).
- Mehadi Hossain and Hossen A Mustafa. 2023. [Automated writing evaluation using sentence by sentence scoring model](#). In *2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM)*, pages 1–6.
- Sungho Jeon and Michael Strube. 2021. [Countering the influence of essay length in neural essay scoring](#). In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 32–38, Virtual. Association for Computational Linguistics.
- Stefan D. Keller, Johanna Fleckenstein, Maleika Krüger, Olaf Köller, and André A. Rupp. 2020. [English writing skills of students in upper secondary education: Results from an empirical study in switzerland and germany](#). *Journal of Second Language Writing*, 48:100700.
- Stefan D. Keller, Julian Lohmann, Ruth Trüb, Johanna Fleckenstein, Jennifer Meyer, Thorben Jansen, and Jens Möller. 2024. [Language quality, content, structure: What analytic ratings tell us about efl writing skills at upper secondary school level in germany and switzerland](#). *Journal of Second Language Writing*, 65:101129.
- Vivekanandan Kumar and David Boulanger. 2020. [Explainable automated essay scoring: Deep learning really has pedagogical value](#). *Frontiers in Education*, 5.
- Vivekanandan S. Kumar and David Boulanger. 2021. [Automated essay scoring and the deep learning black box: How are rubric scores determined?](#) *International Journal of Artificial Intelligence in Education*, 31(3):538–584.
- 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 the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI’19/IAAI’19/EAAI’19*. AAAI Press.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukošiušė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. [Measuring Faithfulness in Chain-of-Thought Reasoning](#). *Preprint*, arXiv:2307.13702.
- Gyeong-Geon Lee, Ehsan Latif, Xuansheng Wu, Ninghao Liu, and Xiaoming Zhai. 2024. [Applying large language models and chain-of-thought for automatic scoring](#). *Computers and Education: Artificial Intelligence*, 6:100213.
- Shengjie Li and Vincent Ng. 2024. [Conundrums in cross-prompt automated essay scoring: Making sense of the state of the art](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7661–7681, Bangkok, Thailand. Association for Computational Linguistics.
- Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. 2022. [Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond](#). *Knowledge and Information Systems*, 64(12):3197–3234.
- Zachary C. Lipton. 2018. [The mythos of model interpretability](#). *Commun. ACM*, 61(10):36–43.
- Julian F. Lohmann, Fynn Junge, Jens Möller, Johanna Fleckenstein, Ruth Trüb, Stefan Keller, Thorben Jansen, and Andrea Horbach. 2024. [Neural networks or linguistic features? - comparing different machine-learning approaches for automated assessment of text quality traits among l1- and l2-learners’ argumentative essays](#). *International Journal of Artificial Intelligence in Education*.
- Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. [Post-hoc interpretability for neural nlp: A survey](#). *ACM Comput. Surv.*, 55(8).
- Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. [Can large language models automatically score proficiency of written essays?](#) In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.
- 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).
- 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.

- 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 Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 316–325, Florence, Italy. Association for Computational Linguistics.
- Christoph Molnar. 2025. *Interpretable Machine Learning*, 3 edition.
- Swapnil Parekh, Yaman Kumar Singla, Changyou Chen, Junyi Jessy Li, and Rajiv Ratn Shah. 2020. [My teacher thinks the world is flat! interpreting automatic essay scoring mechanism](#). *Preprint*, arXiv:2012.13872.
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. [An automated essay scoring systems: a systematic literature review](#). *Artificial Intelligence Review*, 55(3):2495–2527.
- Cynthia Rudin. 2019. [Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead](#). *Nature Machine Intelligence*, 1(5):206–215.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#). *Preprint*, arXiv:1910.01108.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don’t always say what they think: unfaithful explanations in chain-of-thought prompting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA. Curran Associates Inc.
- Felipe Urrutia, Cristian Buc, Roberto Araya, and Valentin Barriere. 2025. [Unsupervised automatic short answer grading and essay scoring: A weakly supervised explainable approach](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 38–54, Vienna, Austria. Association for Computational Linguistics.
- Bronwyn Woods, David Adamson, Shayne Miel, and Elijah Mayfield. 2017. [Formative essay feedback using predictive scoring models](#). In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’17*, page 2071–2080, New York, NY, USA. Association for Computing Machinery.
- Wenbo Xu, Rohana Mahmud, and Wai Lam Hoo. 2024. [A systematic literature review: Are automated essay scoring systems competent in real-life education scenarios?](#) *IEEE Access*, 12:77639–77657.
- 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.
- Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. 2025. [DREsS: Dataset for rubric-based essay scoring on EFL writing](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13439–13454, Vienna, Austria. Association for Computational Linguistics.
- Lui Yoshida. 2025. [Are the Reasoning Models Good at Automated Essay Scoring?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8388–8394, Suzhou, China. Association for Computational Linguistics.

A Appendix: Complementary Figures

A.1 Hyperparameters

Parameter	Value
Batch Size	16
Max Length	512
Epochs	20
Learning Rate	0.005
Loss Function	MSELoss
Early Stopping Patience	5

Table 5: Training hyperparameters for both essay- and sentence-level models.

A.2 Normalisation of Pseudo-Scores

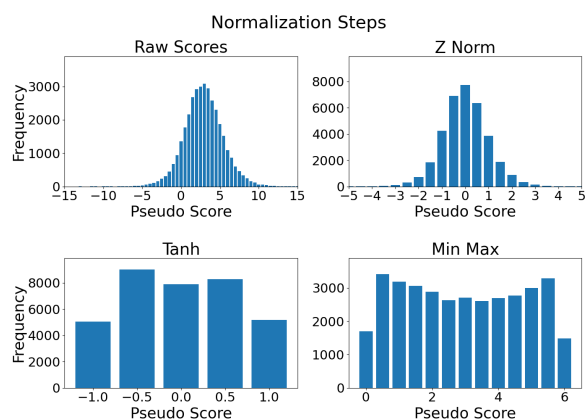


Figure 6: Distribution of sentence pseudo scores across the normalisation steps (z-norm, tanh, min-max) for one particular model.

B Appendix: Sentence-Level Performance

Dataset	Trait	Prompt	Essay	Sent	α	Abl.
MEWS	Content	AD	0.52	0.47	1	1
MEWS	Content	TE	0.45	0.41	1	1
MEWS	Content	TEAD	0.47	0.43	1	1
MEWS	Language	AD	0.70	0.71	1	1
MEWS	Language	TE	0.74	0.75	1	5
MEWS	Language	TEAD	0.72	0.74	1	1
ASAP	Content	1	0.69	0.40	1	1
ASAP	Content	2	0.67	0.61	1	1
ASAP	Content	3	0.60	0.49	1	1
ASAP	Content	4	0.68	0.62	1	5
ASAP	Content	5	0.73	0.70	1	1
ASAP	Content	6	0.69	0.69	1	1
ASAP	Content	7	0.67	0.53	1	1
ASAP	Content	8	0.65	0.24	1	5
ASAP	Language	3	0.57	0.48	1	1
ASAP	Language	4	0.63	0.54	1	5
ASAP	Language	5	0.70	0.54	1	1
ASAP	Language	6	0.61	0.56	1	1

Table 6: Performance (QWK) of essay- and sentence-level linear regression models.

Dataset	Trait	Prompt	Essay	Sent	α	Abl.
MEWS	Content	AD	0.33	0.29	20	1
MEWS	Content	TE	0.18	0.14	10	1
MEWS	Content	TEAD	0.30	0.23	1	3
MEWS	Language	AD	0.56	0.53	10	3
MEWS	Language	TE	0.41	0.58	20	1
MEWS	Language	TEAD	0.68	0.53	1	3
ASAP	Content	1	0.61	0.21	1	3
ASAP	Content	2	0.58	0.43	10	1
ASAP	Content	3	0.52	0.44	1	1
ASAP	Content	4	0.48	0.44	1	1
ASAP	Content	5	0.60	0.34	30	1
ASAP	Content	6	0.70	0.35	10	1
ASAP	Content	7	0.60	0.45	1	5
ASAP	Content	8	0.41	0.15	1	1
ASAP	Language	3	0.50	0.38	30	1
ASAP	Language	4	0.41	0.39	1	1
ASAP	Language	5	0.57	0.25	10	1
ASAP	Language	6	0.57	0.32	10	1

Table 7: Performance (QWK) of essay- and sentence-level LLM models.

Dataset	Trait	Prompt	Essay	Sent	α	Abl.
MEWS	Content	AD	0.44	0.35	1	1
MEWS	Content	TE	0.28	0.31	1	5
MEWS	Content	TEAD	0.33	0.45	1	1
MEWS	Language	AD	0.71	0.74	1	1
MEWS	Language	TE	0.46	0.81	1	5
MEWS	Language	TEAD	0.76	0.75	1	5
ASAP	Content	1	0.66	0.43	1	3
ASAP	Content	2	0.67	0.60	1	5
ASAP	Content	3	0.64	0.50	1	1
ASAP	Content	4	0.68	0.50	1	1
ASAP	Content	5	0.70	0.67	1	5
ASAP	Content	6	0.69	0.68	1	1
ASAP	Content	7	0.60	0.48	1	3
ASAP	Content	8	0.52	0.24	1	3
ASAP	Language	3	0.57	0.44	1	1
ASAP	Language	4	0.61	0.49	1	1
ASAP	Language	5	0.68	0.50	1	1
ASAP	Language	6	0.58	0.52	1	1

Table 8: Performance (QWK) of essay- and sentence-level Hybrid models.

Instruction-Following LLMs for Grammatical Error Correction: Analyzing Neutral-Anchored Instructional Sensitivity Across Editing Modes

Tolgahan Türker¹ and Gülşen Eryiğit²

Faculty of Computer and Informatics

Istanbul Technical University

Istanbul, Türkiye

turkert21@itu.edu.tr¹

gulsen.cebiroglu@itu.edu.tr²

Abstract

Grammatical Error Correction (GEC) requires models to make edit decisions under competing objectives: correcting errors while either minimizing changes or maximizing fluency. However, we lack a principled characterization of how instruction-following Large Language Models (LLMs) shift their edit decisions across such *editing modes*, and whether standard evaluation setups faithfully reflect these shifts. We address this gap by defining three modes—*Neutral*, *Minimal-Edit*, and *Fluency-Edit*—and measuring *neutral-anchored* performance shifts to quantify instructional sensitivity. We benchmark seven LLMs, including proprietary and open-weight models, in a unified zero-shot prompting schema on CoNLL-2014, BEA-2019, and JFLEG datasets. The Minimal-Edit instruction mitigates over-editing and typically boosts precision; in some settings, strong models also improve recall, suggesting more selective and effective corrections. In contrast, the Fluency-Edit instruction often encourages broader paraphrastic rewriting that may improve perceived fluency while lowering GLEU, suggesting both a metric-objective mismatch and a shift away from targeted local correction. Notably, Claude-Sonnet-4.5 demonstrates superior zero-shot capabilities, outperforming previously reported scores and matching or even exceeding few-shot results across CoNLL-2014 ($F_{0.5}$: 67.05), BEA-2019 ($F_{0.5}$: 64.91), and JFLEG ($GLEU$: 66.09).

1 Introduction

GEC is a well-established natural language processing task that involves automatically correcting grammatical errors in text. From a computational perspective, GEC is commonly modeled as a text-to-text transformation problem, where an errorful input is transformed into a corrected output while preserving the author’s intended meaning. Depending on the target application, GEC systems

face distinct requirements: high-quality text generation use cases necessitate fluency-focused revisions on the text to achieve more natural-sounding expression, whereas pedagogical settings often prioritize conservative, minimal edits to gradually introduce learner feedback tailored to the learner’s proficiency level to facilitate a more effective and pedagogically grounded learning experience (Wood et al., 1976; Lyster and Ranta, 1997).

These divergent objectives imply an inherent trade-off: strictly adhering to minimal edits often leaves unnatural or awkward phrasing untouched, whereas aggressively pursuing fluency often leads to broader rewrites that can inadvertently shift the author’s intended meaning. For over a decade, shared tasks and benchmarks for GEC (e.g., CoNLL (Ng et al., 2013, 2014), BEA (Bryant et al., 2019), JFLEG (Napoles et al., 2017), MultiGEC (Masciolini et al., 2025)) have both driven progress and standardized evaluation practices, making the *minimal-edit* and *fluency-edit* distinction central to assessment of GEC systems.

Recent advances in LLMs—particularly their ability to follow complex instructions and produce high-quality outputs—have motivated researchers to investigate their effectiveness for a wide range of tasks, which have traditionally relied on task-specific models developed over many years. In the context of English GEC, a growing body of work (Wu et al., 2023; Fang et al., 2023; Coyne et al., 2023; Loem et al., 2023; Davis et al., 2024) has begun to explore this potential, analyzing the out-of-the-box performance of LLMs under zero- and few-shot settings across standard benchmarks. While these studies provide a valuable baseline, there remains an opportunity to systematically assess contemporary models’ intrinsic capability by examining how their performance shifts across different editing modes, especially when these objectives are specified through structurally consistent instructional framing. To address this, we present an eval-

uation of six contemporary LLMs—ranging from high-capacity systems to efficient mini variants, and including both proprietary and open-weight families—together with a legacy model used to assess the quality of instructions, under a tripartite instructional framework in a zero-shot setting. We define three distinct editing modes: *Neutral*, *Minimal-Edit*, and *Fluency-Edit*, using structurally consistent prompts to observe how models modulate their output in response to different instructional goals. By quantifying the shifts from a *neutral-anchored* baseline, we characterize the instructional sensitivity of these models. To support future benchmarking efforts in the GEC community, our complete experimental framework, including implementation code, instructions, and model-generated outputs, is released for public use.¹

Our primary contributions are: (I) To the best of our knowledge, we achieve the state-of-the-art performances for English prompt-based GEC, with Claude-Sonnet-4.5 achieving the highest reported zero-shot $F_{0.5}$ scores on CoNLL-2014 (67.05) and BEA-2019 (64.91), thereby redefining the zero-shot baseline for the task, (II) we provide empirical evidence for an evolution in instructional sensitivity, contrasting the behavioral rigidity of legacy models like GPT-3.5-Turbo with the higher instructional sensitivity of current frontier models, (III) we characterize the precision-recall (P-R) dynamics across editing modes, demonstrating that while minimal-edit instruction typically acts as a precision filter, it can effectively circumvent the traditional P-R trade-off in top-tier models by aligning generative priors with GEC-specific objectives, (IV) we provide an analysis for the metric-objective mismatch on the JFLEG benchmark, showing that as models exhibit higher instructional sensitivity, their native-like enhancements are penalized by the rigid word-matching constraints of GLEU.

2 Related Works

Initial prompt-based GEC studies primarily focused on the GPT-3.5 family. Wu et al. (2023) compared ChatGPT against specialized tools like Grammarly² and GECTOR (Omelianchuk et al., 2020), noting that its performance on automated metrics was hindered by a tendency to go beyond localized edits to modify surface expressions and

sentence structures. This was further characterized by Fang et al. (2023), who demonstrated that ChatGPT’s human-like fluency often comes at the expense of precision on traditional benchmarks due to its over-correction tendencies. Coyne et al. (2023) extended this analysis to GPT-4, demonstrating a clear performance gap between benchmarks: while the model set a new high score on the fluency-based JFLEG dataset, it significantly underperformed on BEA-2019. Their findings confirm that the GEC performance of LLMs is highly dependent on the targeted edit settings, as prompt-driven fluency often results in extensive revisions that are penalized by minimal-edit metrics.

While initial work established a performance baseline, subsequent research has shifted toward investigating the instructional sensitivity of LLMs and the extent to which their correction behaviors can be strategically modulated. Loem et al. (2023) demonstrated that GPT-3 can be effectively steered toward either *minimal-edit* or *fluency-edit* objectives through instructions. However, their analysis revealed that while instructions can guide the model, the provided examples for few-shot settings play a more critical role in controlling editing behavior and achieving the desired correction style. This sensitivity to prompting is further emphasized by Davis et al. (2024), who conducted an extensive comparison between seven open-source and three commercial LLMs across four benchmarks. They found that while commercial models remain superior in fluency-oriented tasks, open-source models can achieve competitive performance if the prompting strategy is carefully optimized. The requirement for structured, objective-specific instructions is further reinforced by Zeng et al. (2024), showing that a uniform prompting approach is insufficient for addressing the varying error patterns across different language proficiency levels, as over-correction tendencies become more pronounced in advanced learner texts.

A critical challenge in modern GEC is the inherent bias of traditional metrics, where LLMs are often penalized for valid, native-like corrections that deviate from the conservative, minimal edits found in standard references (Östling et al., 2024). To mitigate the metric penalties associated with non-mandatory edits, Staruch et al. (2025) explored methods to steer models toward the principle of least change, effectively constraining the output to adhere to minimal-edit requirements. Alternatively, Liang et al. (2025) introduced edit-wise

¹<https://github.com/tolgahanturker/gec-llm-eval>

²<https://www.grammarly.com>

Neutral	Minimal-Edit	Fluency-Edit
ROLE: You are a helpful language error correction assistant. TASK: Correct the sentence provided inside <code><input>...</input></code> tags. RULES: Process: • Read the full sentence to understand its meaning. • Identify errors in the sentence. • Apply edits. Constraints: • Preserve the original meaning. Output: • If the sentence is already correct, return it unchanged. • Return ONLY the corrected sentence as plain text. • Do NOT include explanations, steps, or additional text.	ROLE: You are a helpful language teaching assistant for second language learners. TASK: Correct the sentence provided inside <code><input>...</input></code> tags with MINIMAL edits. RULES: Process: • Read the full sentence to understand its meaning. • Identify grammatical, spelling, and punctuation errors. • Apply only the edits necessary to fix those errors. Constraints: • Do NOT rewrite for style or fluency. • Preserve the original meaning, tone, and sentence structure as much as possible. Output: • If the sentence is already correct, return it unchanged. • Return ONLY the corrected sentence as plain text. • Do NOT include explanations, steps, or additional text.	ROLE: You are a native English speaker. TASK: Correct grammatical errors in the sentence provided inside <code><input>...</input></code> tags and make it sound natural to a native speaker of English. RULES: Process: • Read the full sentence to understand its meaning. • Identify errors. • Apply edits to correct errors and improve fluency of the sentence. Constraints: • Preserve the original meaning and intent. Output: • If the sentence is already correct, return it unchanged. • Return ONLY the corrected sentence as plain text. • Do NOT include explanations, steps, or additional text.

Table 1: Instructions defining the three editing modes: *Neutral*, *Minimal-Edit*, and *Fluency-Edit*.

preference optimization, an alignment technique that helps models distinguish between mandatory grammatical fixes and optional enhancements.

A different methodological angle pursues automatic optimization of the prompt. Chernodub et al. (2025) proposed APIO, an automatic prompt induction and optimization method that iteratively refines a list of instructions through beam search over an LLM-generated candidate pool, using validation-set performance as the optimization signal.

Building on these efforts to modulate model behavior, our work introduces a structured tripartite instructional framework to systematically characterize how contemporary LLMs navigate these competing objectives of minimal-edit and fluency-edit.

3 Instructions

To isolate the impact of different editing modes, we designed a unified prompting schema following widely accepted best practices for prompt engineering. Each prompt is structured as a consistent interface between the user and the model, consisting of five core components: role framing, task description, processing rules, constraints, and output formatting. This modular design ensures that

model behavior is strictly governed by the intended correction regime while minimizing extraneous output.

We define three editing modes to probe model capabilities: (I) *Neutral*, which serves as an anchor to observe default correction behavior without specific constraints; (II) *Minimal-Edit*, which restricts changes to the smallest set of edits required for correctness while preserving sentence structure; and (III) *Fluency-Edit*, which targets native-like naturalness and flow while only requiring meaning preservation.

All prompts utilize a consistent interface where the source text is encapsulated in `<input>...</input>` tags. To eliminate the need for post-processing on model-generated outputs, we enforce a strict output contract: models are required to return only the corrected sentence, without any additional text. Table 1 shows the instructions we used.

Unlike the other modes, the *Fluency-Edit* instruction explicitly incorporates the term *grammatical errors* in the task to provide a tighter constraint on the model’s editing behavior. This design choice aims to mitigate the common over-editing issue observed in prompt-based GEC studies by guid-

Track	Model	Organization	Access	Inference Platform / API
High-Capacity	GPT-4.1-2025-04-14	OpenAI	Proprietary	OpenAI API
	Claude-Sonnet-4.5	Anthropic	Proprietary	Anthropic API
	Llama-3.3-70B-Instruct	Meta	Open-weight	Azure AI Foundry
	Mistral-Large-3	Mistral AI	Open-weight	Azure AI Foundry
Efficient Mini	GPT-5-mini-2025-08-07	OpenAI	Proprietary	OpenAI API
	Llama-4-Scout-17B-16E-Instruct	Meta	Open-weight	Azure AI Foundry
Legacy	GPT-3.5-Turbo-0125	OpenAI	Proprietary	OpenAI API

Table 2: Summary of evaluated LLMs.

ing the model to prioritize grammatical correctness while striving for native-like flow. To avoid potential conservative bias from corrective roles, we used a *native speaker* persona to prioritize naturalness. As a robustness check, we also tested other personas to see if this choice contributed to the performance decline detailed in Section 5. However, preliminary evaluations—including tests with alternative persona (role) assignments—revealed that these instructional refinements did not alleviate the persistent performance degradation in this mode.

4 Experiments

As listed in Table 2, in our experiments, we evaluate seven models across three tracks to cover various scales and deployment tiers. The High-Capacity track represents the performance upper bound for our experiments, comprising large-capacity commercial models known for superior instruction-following alongside leading open-weight representatives. To reflect the industry trend toward compact intelligence, the Efficient Mini track includes models optimized for low-latency and cost-efficiency. Finally, the Legacy track includes GPT-3.5-Turbo as a historical reference point to measure the quality of the instructions used in the study and the progress made by newer generations in zero-shot GEC.

Experiments are conducted on three standard benchmarks: CoNLL-2014 (Ng et al., 2014), utilizing alternative gold references to better account for the diverse but valid outputs of LLMs in minimal-edit tasks; BEA-2019 (W&I + LOCNESS) (Bryant et al., 2019), providing a large-scale evaluation across the full CEFR spectrum and native student essays; and JFLEG (Napoles et al., 2017), which targets overall naturalness and native-like flow to assess fluency-oriented rewriting capacity.

Our experimental framework is intended to minimize formatting or inference-related inconsistencies, allowing the performance metrics to more

accurately reflect the actual correction capabilities of the models. To do that, we utilize a series of alignment steps that first optimize the source text and then reformat the generated outputs to meet specific benchmark requirements.

Pre-processing. Since the source sentences in the benchmarks are typically provided in tokenized format, we need to detokenize the sentences before LLM processing to provide a more natural linguistic context. Following the practices established by Davis et al. (2024), we employ the *Moses detokenizer* (Koehn et al., 2007) and implement a specific rule to resolve split negative contractions (e.g., *do n’t* $\rightarrow$ *don’t*). We expect this processing to help align the input format with the models’ pre-trained capabilities.

Evaluation Alignment. To facilitate precise scoring, model-generated responses must be re-tokenized to match the specific requirements of each benchmark’s evaluation script. For the BEA-2019 test set, we conduct tokenization via *spaCy*³ to facilitate the creation of .m2 files, directly aligning with the official shared task proposal (Bryant et al., 2019). For CoNLL-2014, we apply the *NLTK tokenizer*⁴ to generate text-based result files, following the conventions suggested in the shared task (Ng et al., 2014). These steps ensure that the model outputs are perfectly aligned with the gold references before being processed by the M^2 and ERRANT scorers.

Inference Details. All evaluations are conducted in a zero-shot setting to assess the out-of-the-box capabilities of the models. To minimize sampling variability, we set the temperature parameter to 0 for all models. Furthermore, to accommodate the varying Tokens Per Minute (TPM) and Requests Per Minute (RPM) constraints of the different API providers, a tailored `delay_per_request` value is implemented for each model. This en-

³<https://spacy.io/>

⁴<https://www.nltk.org/>

sured optimal throughput while strictly adhering to provider-specific rate-limit quotas. The corresponding TPM and RPM values are provided in the GitHub repository. The maximum output length is set to 2048 tokens to avoid overly long generations.

Experimental Design. To align the instructions with each benchmark’s objectives, and to quantify how much model performance shifts from the Neutral prompt (which imposes no specific editing bias), we evaluate CoNLL-2014 and BEA-2019 under Neutral and Minimal-Edit instructions, and JFLEG under Neutral and Fluency-Edit instructions. This design probes whether models can suppress inherent over-editing tendencies under the minimal-edit mode or, conversely, leverage their generative capacity to produce more fluent, natural-sounding output. For this measurement to be interpretable, we restrict our evaluation to a zero-shot setting in order to isolate the effect of instructional framing: introducing in-context demonstrations would conflate instruction-driven behavioral shifts with example-driven imitation, undermining the *neutral-anchored* sensitivity measurements that motivate this work.

5 Results and Analysis

Table 3 summarizes our findings across model scales and editing settings. We categorize the evaluated LLMs into three tracks: High-Capacity, Efficient Mini, and Legacy. For each model, results are reported under three distinct instructions—*Neutral*, *Minimal-Edit*, and *Fluency-Edit*. Following standard practice and recommendations provided in the recent survey by Bryant et al. (2023), we report the $F_{0.5}$ score using the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) for CoNLL-2014, the ERRANT scorer (Bryant et al., 2017) for BEA-2019, and we employ GLEU metric (Napoles et al., 2015) for JFLEG.

To benchmark our results, we include two comparative tiers: (I) Prior LLM Baselines, which present best findings from prior prompt-based LLM evaluations reported in the literature, and (II) single-system state-of-the-art (SOTA) task-specific models.

5.1 Comparative Evaluation

Minimal-Edit Mode. As shown in Table 3, the majority of the evaluated LLMs demonstrate a substantial performance leap over prior prompt-based studies under minimal-edit evaluation settings. In

our experiments, **Claude-Sonnet-4.5** emerges as the top-performing model, achieving the best reported results among prompt-based LLM evaluation studies—presented in the middle part of the table—with an $F_{0.5}$ score of **67.05** on CoNLL-2014 and **64.91** on BEA-2019. Notably, these scores do not merely surpass previous prompt-based GEC studies; they redefine the expected zero-shot baseline for the task.

Results indicate a notable shift in the GEC zero-shot baseline across CoNLL-2014 and BEA-2019. On CoNLL-2014, all evaluated LLMs under the minimal-edit setting outperform prior prompt-based studies, regardless of shot count. Similarly, the majority of high-capacity models surpass previous BEA-2019 prompt-based baselines. While Mistral-Large-3 and Llama-4-Scout trail the 16-shot peak of 57.41, both exceed the zero-shot performance of 53.07 reported by Loem et al. (2023). This suggests that recent open-weight models have reached a level of inherent instruction-following capability where their zero-shot baselines can match or even exceed the few-shot results of earlier models.

A more recent prompt-based comparison point is APIO (Chernodub et al., 2025), which reports a BEA-2019 $F_{0.5}$ of 59.40 using GPT-4o with a validation-tuned, automatically optimized 10-instruction prompt. APIO is methodologically distinct from the few-shot baselines discussed above: rather than supplying in-context demonstrations, it iteratively refines the prompt itself on a validation set drawn from BEA-2019-Dev. Despite this active optimization, three of our four high-capacity models—Claude-Sonnet-4.5 (64.91), GPT-4.1 (64.70), and Llama-3.3-70B (61.61)—surpass APIO under the minimal-edit setting using fixed, manually authored prompts that receive no benchmark-specific tuning. Importantly, a direct comparison between automatic prompt optimization and fixed prompts is not possible from these results, as the underlying models differ. What these results do indicate is that careful zero-shot prompt design, when paired with contemporary high-capacity LLMs from a model class comparable to GPT-4o, surpasses APIO on BEA-2019.

However, when compared to previous single-system SOTA models, a distinct performance disparity emerges between the two benchmarks. While our top-performing LLM, Claude-Sonnet-4.5, achieves highly competitive results on CoNLL-2014 (67.05 $F_{0.5}$, trailing Liang et al. (2025)’s

Track	Model	CoNLL-2014			BEA-2019			JFLEG
		P	R	F _{0.5}	P	R	F _{0.5}	GLEU
High-Capacity	Llama-3.3-70B[†]							
	Neutral	59.45	60.18	59.60	50.14	70.13	53.17	63.68
	Minimal-edit	67.12	53.99	64.01	60.45	66.76	61.61	-
	Fluency-edit	-	-	-	-	-	-	62.04
	Mistral-Large-3[‡]							
	Neutral	49.64	64.06	51.98	35.11	68.84	38.93	59.88
	Minimal-edit	60.15	60.25	60.17	50.46	70.57	53.51	-
	Fluency-edit	-	-	-	-	-	-	50.06
	GPT-4.1[‡]							
	Neutral	57.26	65.84	58.79	46.58	72.54	50.17	65.34
	Minimal-edit	70.37	52.73	65.96	63.75	68.77	64.70	-
	Fluency-edit	-	-	-	-	-	-	60.37
	Claude-Sonnet-4.5[‡]							
	Neutral	64.58	61.71	63.98	57.63	70.67	59.84	66.09
	Minimal-edit	69.14	59.82	67.05	63.48	71.33	64.91	-
	Fluency-edit	-	-	-	-	-	-	63.82
Efficient Mini	Llama-4-Scout[†]							
	Neutral	58.12	60.78	58.63	46.88	69.49	50.15	61.11
	Minimal-edit	63.06	58.17	62.02	53.78	69.11	56.28	-
	Fluency-edit	-	-	-	-	-	-	58.70
	GPT-5-Mini[‡]							
	Neutral	44.33	66.55	47.51	29.77	66.29	33.46	55.12
Legacy	Minimal-edit	62.52	61.34	62.28	54.97	70.52	57.51	-
	Fluency-edit	-	-	-	-	-	-	48.08
	GPT-3.5-Turbo[‡]							
	Neutral	65.05	53.96	62.48	54.70	65.97	56.63	64.67
	Minimal-edit	65.93	53.65	63.05	54.43	65.91	56.39	-
	Fluency-edit	-	-	-	-	-	-	63.99
Prior LLM Baselines								
Fang et al. (2023)	GPT-3.5-Turbo [§]	51.3 ³	62.4 ³	53.2 ³	34.6 ¹	69.7 ¹	38.4 ¹	63.5 ³
Coyne et al. (2023)	GPT-3.5 td-003*	-	-	-	-	-	49.66 ²	63.40 ²
Coyne et al. (2023)	GPT-4-0314	-	-	-	-	-	52.79 ²	65.02 ²
Loem et al. (2023)	GPT-3.5 td-003*	-	-	57.06 ¹⁶	-	-	57.41¹⁶	69.25⁶⁴
Davis et al. (2024)	Stable Beluga 2	-	-	57.2 ⁰	-	-	-	61.3 ⁰
Davis et al. (2024)	GPT-3.5-Turbo-0613	-	-	57.2 ¹	-	-	-	62.5 ⁴
Omelianchuk et al. (2024)	GPT-4-0613	59.0 ⁰	55.4 ⁰	58.2⁰	-	-	-	-
Chernodub et al. (2025)	GPT-4o-2024-05-13	-	-	-	-	-	59.40^{opt}	-
Single-System SOTA								
Stahlberg and Kumar (2021)	Seq2Seq Transformer	72.8	49.5	66.6	72.1	64.4	70.4	64.7
Liang et al. (2025)	Mistral-7b-EPO	76.71	52.56	70.26	78.16	68.07	75.91	-
Staruch et al. (2025)	Gemma 2 (27b) TS	77.38	47.88	68.89	82.28	67.03	78.70	62.42

Table 3: GEC performance across different model families under three editing modes. Superscripts denote the number of few-shot samples. [†]: Open-weight, [‡]: Proprietary, [§]: Snapshot unspecified, td-003*: text-davinci-003, ^{opt}: Optimized prompt with induced instructions, not few-shot. Model names in the top part are abbreviated for clarity; full model identifiers can be found in Table 2.

70.26 by 3.21 points), a clear performance gap remains on BEA-2019. Even under the minimal-edit setting, the best zero-shot LLM performance (64.91) still trails behind the current empirical upper bound of 78.70 (Staruch et al., 2025), highlighting the continued necessity of task-specific solutions for high-complexity GEC datasets.

This performance gap—rather than indicating a

fundamental lack of GEC capability in LLMs—highlights the effectiveness of domain-specific training and benchmark-centric optimization. Specialized models benefit from exposure to the specific correction styles and error distributions inherent in the benchmark training sets. This exposure allows them to internalize benchmark-specific repair patterns—patterns that are not accessible

to general-purpose LLMs in a zero-shot setting, where the only available signal is the instructional framing and task constraints provided in the prompt.

Fluency-edit Mode. The results on the JFLEG benchmark reveal a counter-intuitive relationship between fluency constraint and model performance. As shown in Table 3, **Claude-Sonnet-4.5** again emerges as the leader among the evaluated models, achieving a peak GLEU score of **66.09** under the neutral setting. Interestingly, across all model tracks, the fluency-edit instruction leads to a slight but consistent performance degradation compared to the neutral prompt. This observation suggests a potential overshooting effect: when explicitly tasked with maximizing fluency while maintaining adherence to grammatical correctness, LLMs can be driven toward excessive paraphrasing that diverges from the gold-standard references. A discussion of this phenomenon is provided in Appendix A.

Claude-Sonnet-4.5, GPT-4.1, and GPT-3.5-Turbo surpass prior zero-shot baselines. Although trailing the 64-shot peak (69.25 GLEU) of [Loem et al. \(2023\)](#), our zero-shot top result (66.09) exceeds their zero-shot baseline of 64.51. Notably, GPT-3.5-Turbo outperforms Efficient Mini models, suggesting smaller contemporary architectures still trail their larger predecessors on this benchmark.

Compared to previous SOTA results, the fluency performance of most of the LLMs is remarkably competitive. Claude-Sonnet-4.5 and GPT-4.1 both surpass the specialized model baseline of 64.7 reported by [Stahlberg and Kumar \(2021\)](#). Most of the high-capacity models and even the legacy GPT-3.5-Turbo outperform the recent task-specific model ([Staruch et al., 2025](#)) on this benchmark. This suggests that for fluency-centric GEC, the broad linguistic knowledge of general-purpose LLMs is now reaching a level where they can exceed the performance of models specifically tuned for error correction, as long as the prompt maintains a balance between fluency correction and original structure.

To further investigate the linguistic drivers behind the consistent performance decline observed in the Fluency-Edit setting, we conduct a qualitative case study on a representative subset of models. This analysis reveals that the lower scores often stem from a metric-objective mismatch rather than a failure in instruction adherence. While the

Fluency-Edit prompt successfully triggers more naturalizing, global paraphrastic reformulations, these high-quality enhancements are frequently penalized for their structural divergence from the rigid n-gram boundaries of the gold references. We provide a granular discussion of these behavioral shifts and model-specific failure cases in Appendix A.

5.2 Prompt Quality through Legacy Benchmarking

To assess the effectiveness of instructions used in this study, we conduct a comparative analysis using GPT-3.5-Turbo as a reference point against prior prompt-based studies. We acknowledge that an exact one-to-one comparison is hindered by the temporal nature of proprietary model snapshots; while earlier studies utilized versions such as gpt-3.5-turbo-0613 or text-davinci-003, our experiments employ the 0125 snapshot, which was the most recent available version at the time of this study.

As shown in Table 3, our minimal-edit prompt yields a substantial performance gain on GPT-3.5-Turbo compared to previous scores reported for GPT-3.5 family. On CoNLL-2014, our zero-shot configuration achieves an $F_{0.5}$ of 63.05, which not only beats the zero-shot baselines but also exceeds every few-shot score recorded in prompt-based GEC studies to date. Remarkably, this performance even surpasses the results reported for subsequent and more advanced iterations; for instance, our zero-shot GPT-3.5-Turbo score significantly outperforms the 58.2 $F_{0.5}$ achieved by GPT-4-0613 in [Omelianchuk et al. \(2024\)](#). This trend is even more pronounced on BEA-2019, where it reaches 56.63 and 56.39 $F_{0.5}$ in neutral and minimal-edit settings respectively, nearly rivaling the 16-shot performance (57.41) of [Loem et al. \(2023\)](#) and outperforming [Fang et al. \(2023\)](#)’s 38.4 and even [Coyne et al. \(2023\)](#)’s 52.79 with GPT-4.

Regarding the evaluation on JFLEG, GPT-3.5-Turbo GLEU results for both neutral (64.67) and fluency-edit (63.99) settings outperforms all GPT-3.5 family zero-shot and low-shot baselines reported in prior work ([Fang et al., 2023](#); [Coyne et al., 2023](#); [Davis et al., 2024](#)). Although [Loem et al. \(2023\)](#) achieves a higher score with an extensive 64-shot setup (69.25), our zero-shot performance for GPT-3.5-Turbo surpasses their zero-shot baseline of 64.51.

Collectively, these results demonstrate that the primary driver of performance in prompt-based GEC is not merely the raw parameter count of the

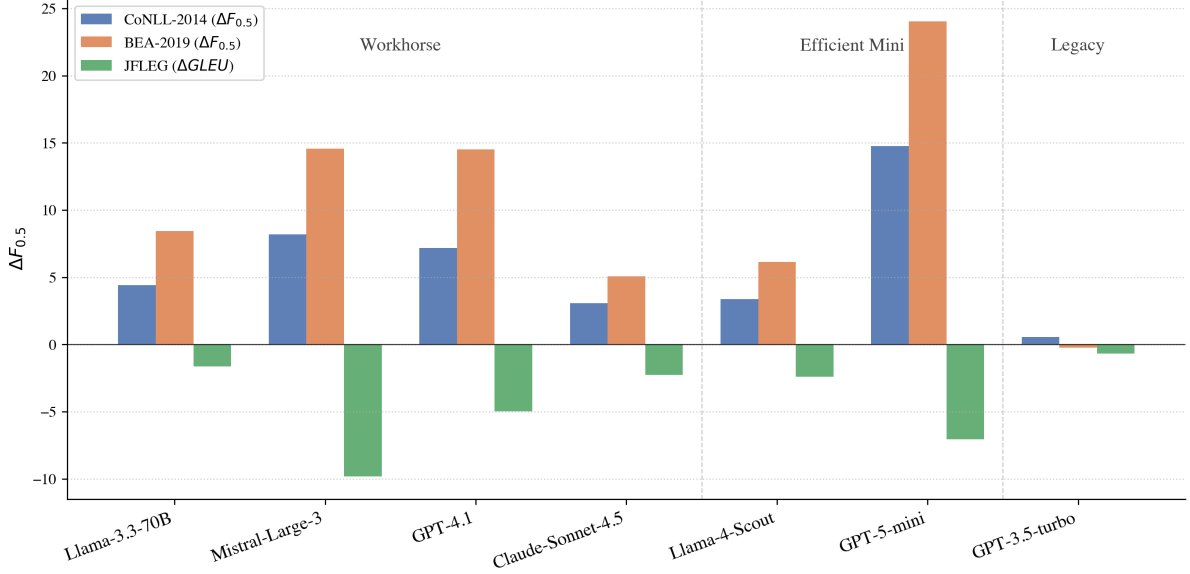


Figure 1: Delta changes in performance when switching from the Neutral baseline to Minimal-Edit or Fluency-Edit

underlying model, but the clarity of the task framing and instructional specificity of the prompts. We argue that the consistent superiority of our prompts—even over significantly larger models like GPT-4—stems from their ability to navigate the correction-preservation trade-off with higher sensitivity. In the minimal-edit configuration, our instructions impose rigorous decision boundaries that successfully inhibit the over-editing bias common in LLMs, allowing the model to prioritize precision over aggressive rewriting. Conversely, in the fluency setting, score divergence reflects a shift toward paraphrastic reformulation rather than instructional failure. This overshooting effect triggers a metric-objective mismatch, where idiomatic enhancements are penalized by rigid n-gram boundaries of the metric (Appendix A).

5.3 Instructional Sensitivity and P-R Dynamics

Editing mode transitions reveal distinct behavioral shifts. $F_{0.5}$ score deltas (Figure 1) highlight P-R dynamics, offering insights into a model’s instructional sensitivity. Given that $F_{0.5}$ weights precision twice as much as recall, these performance shifts (deltas) reflect models’ capacity to refine their output under restrictive constraints.

Minimal-Edit instruction consistently improves $F_{0.5}$ scores over the Neutral baseline, a gain primarily driven by sharp increases in Precision as can be seen in Table 3. By explicitly requiring models to preserve original structure and apply minimal

changes, we effectively mitigate the common zero-shot pitfall of over-editing. A notable exception is the legacy GPT-3.5-Turbo. It exhibits behavioral rigidity, with minimal performance deltas indicating a limited response to varying instructional objectives and constraints. This implies that the model is reaching a performance ceiling under the neutral setting and is leaving little headroom for further optimization through explicit instructional objectives.

One of the important trends observed in Table 3 is the significant boost in Precision at the expense of Recall when switching to the Minimal-Edit mode. GPT-4.1 serves as a textbook example of this dynamic on CoNLL-2014, where its Precision jumps from 57.26 to 70.37 (+13.11), while its Recall undergoes an equivalent drop from 65.84 to 52.73 (−13.11). This suggests that for certain models, the Minimal-Edit instruction acts primarily as a conservative filter, successfully suppressing the inherent tendency to perform excessive rewrites. However, for Claude-Sonnet-4.5, GPT-5-mini, and Mistral-Large-3, the Minimal-Edit instruction effectively circumvents this traditional P-R trade-off on the BEA-2019 dataset. Unlike the typical inverse relationship, these models exhibit simultaneous gains in both metrics. This phenomenon indicates that the instruction does not merely impose a restrictive bottleneck; rather, it aligns the model’s generative priors with given GEC-specific objectives. By providing a clearer operational boundary, the prompt enables these models to target actual

errors more accurately while inherently reducing the noise generated by non-corrective rewriting.

Claude-Sonnet-4.5 exhibits a unique performance profile, maintaining the highest $F_{0.5}$ scores with significantly narrower deltas than its peers (Figure 1). Unlike other models that require explicit minimal-edit constraints to mitigate over-editing, Claude’s strong neutral performance suggests an intrinsic minimalist prior. This out-of-the-box maturity indicates the model is relatively better aligned for conservative GEC, bypassing the need for the explicit guidance required by other models.

5.4 Error Type Analysis

The aggregate $F_{0.5}$ shifts reported in Section 5.3 characterize *how much* model behavior changes between editing modes, but they leave the precision-recall mechanics underspecified at the linguistic level. To address this, we conduct an ERRANT-based error-type breakdown for our top-performing model, Claude-Sonnet-4.5, on the BEA-2019 test set under both neutral and minimal-edit instructions. Table 5 (Appendix B) reports precision, recall, and $F_{0.5}$ for the 15 most frequent error categories, which jointly cover approximately 77.27% of the gold-reference edits enumerated by ERRANT for this evaluation.

The breakdown reveals that the aggregate minimal-edit benefit decomposes into three distinct behavioral patterns: rewrite suppression in open-ended substitution categories such as R:VERB and R:OTHER; simultaneous precision and recall gains in rule-governed morphosyntactic categories such as R:MORPH and R:VERB:SVA; and minimal change in already-saturated categories such as R:SPELL. The full breakdown and accompanying analysis are presented in Appendix B.

6 Conclusion

In this study, we evaluated six contemporary LLMs to analyze their performance in GEC. We tested these models under three different editing modes using *Neutral*, *Minimal-Edit*, and *Fluency-Edit* instructions to analyze their responses.

Claude-Sonnet-4.5 set a new performance record for prompt-based GEC, with its zero-shot results outperforming even previously reported few-shot scores. Using a minimal-edit instruction helps improve precision and overall $F_{0.5}$ scores across various model scales. This approach reduces the tendency of models to edit sentences more than

necessary. Furthermore, we found that switching to the Fluency-Edit instruction led to a consistent drop in GLEU scores. This is because models perform rewrites that are mostly correct but do not match the exact words in human references, leading to penalties from the n-gram-based metric. Our results show a clear difference in how models follow instructions. The legacy GPT-3.5-Turbo is markedly less sensitive to editing modes, exhibiting consistently smaller *neutral-anchored* shifts than contemporary LLMs. We argue that the design of the prompt is as critical as the size of the model for effective grammar correction. While general LLMs are becoming competitive with specialized systems, future evaluation methods must account for the fact that GLEU may not accurately capture the quality of fluency improvements.

7 Limitations

We acknowledge some constraints in our study that should be considered. First, while we attribute the performance drop in Fluency-Edit mode to a metric-objective mismatch, this claim remains to be validated through large-scale human evaluation to confirm if LLM-generated rewrites are preferred over gold standards. Second, our experimental design intentionally restricts evaluation to a zero-shot setting in order to isolate instructional sensitivity from in-context example effects. Although few-shot prompting is known to improve absolute performance on GEC, augmenting our best configuration with in-context demonstrations was beyond the scope of this study and is left as a complementary direction for the broader research community. Third, recent work shows that LLM behavior can be sensitive to prompt formatting variations (e.g., delimiter choice, instruction ordering, or layout conventions) (Sclar et al., 2024); our study, which adopts a single fixed structural template, does not address this dimension, and the format-level robustness of editing-mode shifts remains an open empirical question. Fourth, our error-type analysis characterizes instructional sensitivity within zero-shot settings but does not provide a direct head-to-head comparison with fine-tuned specialized systems at the category level. Such a comparison is currently constrained by the absence of category-level breakdowns in published fine-tuned baselines and remains unexplored. Finally, our English-only focus limits generalizability to morphologically rich or low-resource languages.

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*, 49(3):643–701.
- Artem Chernodub, Aman Saini, Yejin Huh, Vivek Kulkarni, and Vipul Raheja. 2025. [APIO: Automatic prompt induction and optimization for grammatical error correction and text simplification](#). In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 234–239, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.
- 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"](#). *Preprint*, arXiv: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.
- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipour, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. 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*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. [Moses: Open source toolkit for statistical machine translation](#). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. 2025. [Edit-wise preference optimization for grammatical error correction](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3401–3414, Abu Dhabi, UAE. 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. 4. 07/2023 (BEA 2023).
- Roy Lyster and Leila Ranta. 1997. Corrective feedback and learner uptake: Negotiation of form in communicative classrooms. *Studies in second language acquisition*, 19(1):37–66.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfah, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025. [The MultiGEC-2025 shared task on multilingual grammatical error correction at NLP4CALL](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 1–33, Tallinn, Estonia. University of Tartu Library.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. [Ground truth for grammatical error correction metrics](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A fluency corpus and benchmark for grammatical error correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 shared task on grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

- 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 Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashnyi. 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.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhashnyi, Artem Chernodub, Oleksandr Kornienko, and Igor Samokhin. 2024. [Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.
- Robert Östling, Katarina Gillholm, Murathan Kurfali, Marie Mattson, and Mats Wirén. 2024. [Evaluation of really good grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6582–6593, Torino, Italia. ELRA and ICCL.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. [Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting](#). In *International Conference on Learning Representations*, volume 2024, pages 25055–25083.
- 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.
- Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. 2025. [Adapting LLMs for minimal-edit grammatical error correction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 118–128, Vienna, Austria. Association for Computational Linguistics.
- 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.
- Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. [ChatGPT or grammarly? evaluating chatgpt on grammatical error correction benchmark](#). *Preprint*, arXiv:2303.13648.
- Min Zeng, Jiexin Kuang, Mengyang Qiu, Jayoung Song, and Jungyeul Park. 2024. [Evaluating prompting strategies for grammatical error correction based on language proficiency](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6426–6430, Torino, Italia. ELRA and ICCL.

A Qualitative Analysis

While the quantitative results provide an empirical foundation, they do not fully capture the nuanced linguistic modifications of the selected LLMs. In particular, the Fluency-Edit instruction on the JFLEG benchmark yields a consistent performance decline as presented in Table 3. To clarify this discrepancy, we examine the relationship between model outputs and metric responses through a representative case study (Table 4).

For this analysis, we select three models that represent distinct behavioral profiles based on the quantitative results: Claude-Sonnet-4.5 as the overall performance leader, Mistral-Large-3 for its high instructional sensitivity (max Δ), and GPT-3.5-Turbo as a representative of behavioral rigidity (min Δ).

Table 4 provides a representative illustration of the broader quantitative trends discussed earlier, serving as a clear instance of the model-specific behavioral patterns identified in previous subsections. For each model, the table displays the generated outputs under both Neutral and Fluency-Edit instructions, alongside their respective sentence-level GLEU scores which are evaluated for each hypothesis against the set of four gold references.

Claude-Sonnet-4.5 is particularly informative because it is our strongest overall model, and its outputs reveal a subtle but important distinction between reference proximity and perceived fluency. Under the Neutral setting, it corrects the clear grammatical error *didn’t expected* to *didn’t expect* and also repairs *were flock to* to *were flocking to*, but otherwise remains highly conservative. Notably, its output remains very close to Ref3, but the omission of the infinitival marker *to* after *expect* leaves the *didn’t expect* construction slightly incomplete and somewhat less natural from a native-speaker perspective. Despite this residual awkwardness, the output receives a relatively strong sentence-level GLEU score (0.69), which is consistent with the metric’s sensitivity to n-gram overlap across the available references rather than to fine-grained judgments of naturalness. Under the Fluency-Edit

Original Sentence:

As a result, she could get a job that she badly wanted but she didn't expected because many people were flock to this company.

Gold References:

Ref0: As a result, she was able to get a job that she badly wanted but didn't expect because so many people were flocking to this company.

Ref1: As a result, she could get a job that she badly wanted but she didn't expect to get because many people flocked to this company.

Ref2: As a result, she could get a job that she badly wanted but didn't expect to get because many people were flocking to this company.

Ref3: As a result, she could get a job that she badly wanted but she didn't expect to because many people were flocking to this company.

Model	Mode	Model Output	GLEU
Claude-4.5	Neutral	As a result, she could get a job that she badly wanted but she didn't expect because many people were flocking to this company.	0.69
	Fluency	As a result, she got a job that she badly wanted but didn't expect because many people were flocking to this company.	0.62
Mistral-L3	Neutral	As a result, she could get a job that she badly wanted but didn't expect because many people were flocking to this company.	0.64
	Fluency	As a result, she got the job she badly wanted, though she hadn't expected it because so many people were flocking to the company.	0.29
GPT-3.5	Neutral	As a result, she could get a job that she badly wanted but she didn't expect because many people were flocking to this company.	0.69
	Fluency	As a result, she could get a job that she badly wanted but she didn't expect because many people were flocking to this company.	0.69

Table 4: Comparison of outputs of the models for the Sentence ID 217 in the JFLEG test set with sentence level GLEU scores.

setting, it makes a more ambitious change by replacing *could get* with *got*, which moves the sentence toward a more natural and direct formulation. However, it still leaves the construction *but didn't expect* unresolved, rather than producing the more complete and idiomatic *didn't expect to* or *didn't expect to get it*. In other words, the fluency-oriented instruction does trigger a more naturalizing edit, but it does not fully repair the most awkward local structure. This makes Claude's pair of outputs especially useful for illustrating that higher fluency intent does not automatically translate into a uniformly better correction, while higher GLEU under the neutral prompt may partly reflect reference closeness rather than superior overall well-formedness.

Mistral-Large-3 exhibits an even clearer contrast between conservative correction and fluency-driven editing. Under the Neutral setting, its output is nearly identical to Claude's neutral response: it corrects the obvious grammatical errors but preserves most of the original sentence structure, yielding a relatively strong GLEU score (0.64). This suggests that, even without explicit editing guidance, the model already performs a balanced correction that lies somewhere between minimal editing and moderate fluency improvement. Under the Fluency-

Edit setting, however, it shifts much more decisively into reformulation: *could get a job* becomes *got the job*, but becomes *though*, *didn't expect* becomes *hadn't expected it*, *many people were flocking to this company* becomes *so many people were flocking to the company*. These revisions make the output more globally rewritten and, at first glance, more polished. At the same time, they substantially increase lexical and structural distance from the references, which likely explains the sharp drop in sentence-level GLEU (0.29). More broadly, this example is consistent with the general qualitative findings: among the evaluated models, Mistral-Large-3 exhibits the largest behavioral shift, making it the clearest case of high instructional sensitivity. Accordingly, the drop in GLEU reflects not only the behavior of the metric, but also a broader change in the model's response to the Fluency-Edit prompt. This is partly consistent with a metric-objective mismatch, insofar as the prompt encourages native-like reformulation while the metric still rewards proximity to a limited reference set. At the same time, the example indicates that the lower score cannot be explained by the metric alone. Some of Mistral's changes are plausibly motivated by fluency, but they also extend the revision beyond targeted local repair. Overall, the Fluency-Edit

prompt shifts the model from balanced correction toward broader paraphrastic rewriting, reducing reference alignment without delivering a clearly proportional gain in local correction quality.

GPT-3.5-Turbo shows a different pattern, remaining unchanged across prompts. Its Neutral and Fluency-Edit outputs are identical, indicating that the fluency instruction has little or no effect on its correction behavior in this example. This is also in line with the general quantitative results: among the evaluated models, GPT-3.5-Turbo shows the smallest change, consistent with its overall profile of low instructional sensitivity. At the sentence level, this low sensitivity is reflected in a restrained editing strategy: the model fixes the obvious grammatical errors but leaves the larger phrasing intact, including the somewhat incomplete *she didn't expect* segment. This yields a relatively high GLEU score (0.69), likely because the output remains reference-close and avoids broader paraphrastic divergence. One interpretation is that GPT-3.5, as a legacy model, is comparatively rigid in following higher-level stylistic instructions such as *make it more fluent* or *more native-like*, especially when these instructions compete with a simpler default strategy of surface-level error correction. This contrasts with Mistral's behavior: whereas Mistral shifts toward more aggressive rewriting under the fluency-edit prompt, GPT-3.5 remains largely unchanged in how it edits. In this sense, the example also provides a concrete instance of the broader behavioral rigidity observed in the quantitative results.

B Error Type Analysis

This appendix complements the Error Type Analysis introduced in Section 5.4 by presenting the full ERRANT-based breakdown and a regime-by-regime discussion of the observed shifts. Table 5 presents a per-category breakdown for the 15 most frequent ERRANT categories, reporting their gold-edit frequency, precision, recall, and $F_{0.5}$ under neutral and minimal-edit instructions, together with the $\Delta F_{0.5}$ between the two modes.

Of the 15 categories examined, 14 exhibit a positive $\Delta F_{0.5}$ when switching from neutral to minimal-edit, with a mean gain of +2.95 points. Crucially, this improvement is not driven by a uniform precision boost across the board: while the average precision increases by +3.70 points, the average recall stays nearly flat (−0.13). The category-

level distribution, however, hides three distinct behavioral profiles that together explain the P-R dynamics observed in Section 5.3.

The largest $F_{0.5}$ gains concentrate in open-ended substitution categories—R:VERB, R:PREP, and R:OTHER—which often involve multiple plausible alternatives or contextually optional changes, and where the minimal-edit prompt produces large precision improvements alongside stable or declining recall, indicating that a substantial portion of the neutral baseline's edits in these categories are filtered out under the minimal-edit constraint. R:VERB ($\Delta F_{0.5}=+8.68$) is the clearest case: precision jumps by +13.75 points while recall drops by −7.87, indicating that the minimal-edit instruction sharply suppresses non-mandatory verb substitutions. R:OTHER ($\Delta F_{0.5}=+5.97$; $\Delta P=+7.35$, $\Delta R=-1.52$) and R:PREP ($\Delta F_{0.5}=+5.25$; $\Delta P=+6.53$, $\Delta R=-1.03$) follow a similar pattern. This pattern provides a category-level localization of the “over-editing” tendency identified in prior prompt-based GEC studies (Wu et al., 2023; Fang et al., 2023), and our results indicate that the minimal-edit mode substantially suppresses it in these categories.

A second, distinct group of categories shows simultaneous gains in both precision and recall. R:MORPH ($\Delta P=+9.14$, $\Delta R=+1.77$), R:VERB:FORM ($\Delta P=+2.98$, $\Delta R=+2.69$), R:VERB:SVA ($\Delta P=+2.40$, $\Delta R=+1.86$), and R:NOUN:NUM ($\Delta P=+3.33$, $\Delta R=+1.39$) all improve on both axes. These categories share a common property: they are governed by relatively deterministic grammatical rules—subject-verb agreement, morphological inflection, number marking—rather than admitting multiple plausible alternatives. For such categories, the minimal-edit instruction yields gains beyond what a purely restrictive filter would predict. A filter operating on neutral outputs could remove false positives and raise precision, but it could not introduce true positives that the neutral run failed to produce. The simultaneous improvement therefore suggests that the prompt influences the correction process itself rather than acting only as a post-hoc constraint, consistent with the observation in Section 5.3 that in top-tier models the minimal-edit prompt may not simply impose a restrictive bottleneck.

A third pattern emerges in error categories where neutral performance is already near ceiling, leaving little headroom for the minimal-edit instruc-

Error Type	Freq.	Neutral			Minimal-Edit			$\Delta F_{0.5}$
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	
M:PUNCT	761	64.56	61.98	64.03	64.90	66.10	65.14	+1.11
R:OTHER	629	34.43	44.76	36.10	41.78	43.24	42.07	+5.97
R:ORTH	464	56.36	68.45	58.42	57.29	71.12	59.61	+1.19
R:SPELL	343	87.79	96.85	89.47	88.36	97.38	90.03	+0.56
R:PREP	340	65.86	76.62	67.76	72.39	75.59	73.01	+5.25
R:NOUN:NUM	313	74.26	87.11	76.52	77.59	88.50	79.55	+3.03
M:DET	306	78.75	75.08	77.98	79.80	77.45	79.32	+1.34
U:DET	257	71.09	83.68	73.30	74.63	77.82	75.24	+1.94
R:VERB:TENSE	223	64.00	68.97	64.94	67.56	68.16	67.68	+2.74
R:VERB	214	52.71	57.87	53.67	66.46	50.00	62.35	+8.68
R:VERB:FORM	200	76.11	84.31	77.62	79.09	87.00	80.56	+2.94
R:MORPH	193	64.02	85.79	67.44	73.16	87.56	75.65	+8.21
R:VERB:SVA	167	75.38	90.36	77.96	77.78	92.22	80.29	+2.33
R:PUNCT	161	36.19	46.34	37.85	32.92	49.69	35.30	-2.55
R:DET	134	67.11	71.33	67.91	70.40	65.67	69.40	+1.49

Table 5: ERRANT error-type breakdown for Claude-Sonnet-4.5 on BEA-2019 under Neutral and Minimal-Edit instructions. We report the 15 most frequent error categories, which together account for approximately 77.27% of all gold edits in the test set. Frequency is computed as TP + FN from the Minimal-Edit run; the Neutral counterpart yields a near-identical ranking, with minor count differences arising from ERRANT’s hypothesis-conditioned alignment. $\Delta F_{0.5}$ denotes the absolute change in $F_{0.5}$ from the Neutral to the Minimal-Edit instruction.

tion to act on. R:SPELL improves only marginally ($\Delta F_{0.5}=+0.56$) because it operates close to a saturation point with both precision (87.79%) and recall (96.85%) already very high under neutral mode. R:ORTH ($\Delta=+1.19$) and M:PUNCT ($\Delta=+1.11$) behave similarly. The instructional framing has limited influence on phenomena whose correction is largely deterministic and where model behavior is already well-calibrated.

R:PUNCT is the only category in the top-15 that shows a negative $\Delta F_{0.5}$ (-2.55), driven by a precision drop (-3.27) accompanied by a recall gain ($+3.35$). This pattern runs counter to the dominant trend and likely reflects a category-specific interaction. Under the minimal-edit mode, the model attempts more punctuation replacements, but its replacement choices are not sufficiently selective. Replacement punctuation is intrinsically context-dependent, with multiple plausible alternatives often available for the same source position. The observed precision drop may reflect the difficulty of aligning model choices with the specific replacements chosen by reference annotators, though we cannot rule out a genuine selectivity issue without examining individual cases.

Assessing the Quality and Consistency of Automated Knowledge Component Generation using Instructor-generated Questions and LLMs

Jordan Esiason¹, Priyanka Khare¹, Wookhee Min¹, Seung Lee¹,
Gamze Ozogul², Xiaoying Zheng², Yeil Jeong², James Lester¹

¹North Carolina State University, ²Indiana University – Bloomington

Correspondence: jesiaso@ncsu.edu

Abstract

Lecture-style instruction is one of the most prevalent forms of learning in postsecondary education in the United States. Despite the factors that make lectures a convenient format, they tend to present few opportunities for meaningful engagement between students and the course materials being presented due to factors such as the overhead associated with interacting with large numbers of students. By utilizing large language models, we have created a pipeline built upon the EXPLAINIT classroom response system for processing student self-explanations produced during lectures using automatically generated knowledge components. This pipeline can facilitate deeper engagement with course materials, offer traceability in assessment results, and allows instructors to respond to student errors or misconceptions in real-time during lecture. While previous work using a proprietary large language model has examined the basic functionality of this pipeline, this work more closely examines the consistency and quality of this pipeline using both a large closed-weight model and a smaller open-weight model, with or without retrieval-augmented generation (RAG). The use of open-source models could allow institutions deploying EXPLAINIT to maintain control of their student data without substantially sacrificing performance. We find that while there are small statistically significant differences in performance between the RAG conditions of each LLM, they are nearly comparable at this task. Additionally, the LLM-generated knowledge components are of higher quality when relevant course material is provided for RAG, although consistency is not improved. These results indicate that both large closed-weight and smaller open-weight models show promise in this task, but fine-tuning may be necessary to improve performance further.

1 Introduction

Lecture-style instruction is one of the most popular teaching formats in the United States due to factors such as course size and the availability of appropriate classrooms (Stains et al., 2018). However, this style of lecture suffers from drawbacks such as a lack of interactivity that can lead to poorer learning outcomes compared to students in more interactive (or dialogic) learning environments (Grissom et al., 2017). Classroom response systems (CRSs) may represent a path to improving student engagement by allowing instructors and students to interact during lectures through question and answer sessions mediated by laptops, handheld devices, or other technological devices, but many CRSs that rely on less engaging question types such as multiple-choice responses may have low or negligible impacts on learning gains (Hunsu et al., 2016). A CRS implementing self-explanation in a real-time lecture format may represent a path to substantially improving engagement and learning gains. While automated assessment of written responses is an active area of study (Gao et al., 2023; Urrutia et al., 2025; Bexte et al., 2024), it still represents a substantial computing challenge, particularly in the context of real-time assessment during lectures. Large language models (LLMs) provide a promising method of explainable automated assessment but require further evaluation with regards to their consistency and correctness. Additionally, privacy concerns such as attack vulnerabilities and data governance have arisen around the use of large private closed-weight LLMs (Li et al., 2024). Small, open-weight models can enable institutions to have total control over their implementations.

In this paper, we expand upon a previously developed explainable LLM-powered pipeline (Esiason et al., 2026a) used to generate knowledge component (KC) sets from a variety of instructor-generated questions, tag student responses with

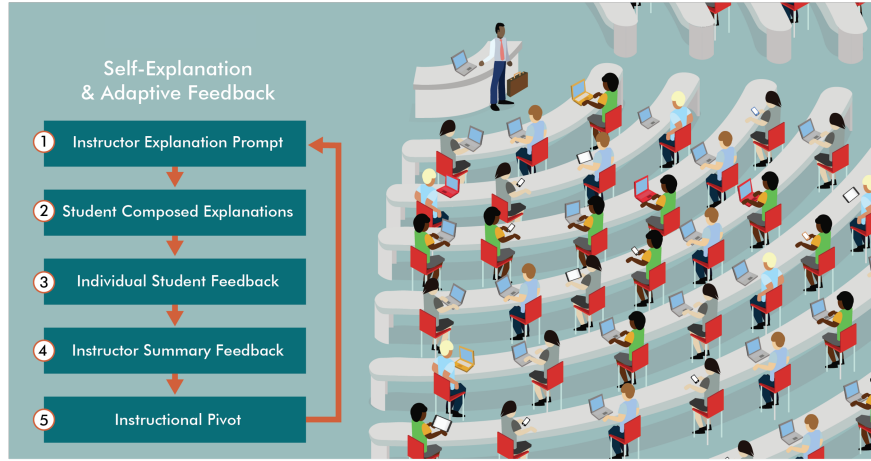


Figure 1: The EXPLAINIT CRS. Instructors are able to use the ExplainIt interface to pose self-explanations questions during lecture, and students can respond to those questions in their own words using a student interface on their handheld devices or laptops. Students receive LLM-generated feedback on their answer and instructors receive a class-level summary of student performance on the question.

these KCs, and summarize topic-level classroom performance by testing a popular closed-weight model (ChatGPT GPT-4o-mini) against a relatively lightweight open-weight model (MistralAI Mistral-3-3B-Instruct). In this work, we explore how consistently KCs are generated when the same instructor questions are used as input across multiple generations of KC sets. We also qualitatively evaluate how RAG utilizing course materials affects KC generation by examining whether or not generated KCs are within the scope of the intent of the question being posed, and whether or not the generated KCs are sufficient in capturing a “correct” response from a student.

Specifically, we seek to evaluate the following research questions:

- RQ1: Does including course materials as RAG context for these models result in the generation of higher quality KCs as measured by the scope of the generated content and the sufficiency of the generated KCs in capturing correct responses?
- RQ2: How consistent are the output of these models over multiple generation cycles as measured by the proportion of times that semantically similar KCs are generated across all generations given the same input? Does consistency vary by the type of question (e.g. factual recall, procedural knowledge, etc.)?
- RQ3: What, if any, other qualitative or quantitative patterns appear in generated KCs with regards to their quality?

By addressing these research questions, we contribute to the understanding of how these two models qualitatively behave in this novel application context, whether or not they are capable of generating useful KCs consistently, and whether or not closed- and open-weight model performance is comparable. Additionally, the human evaluation of this work establishes an expert dataset of KCs that can be used for future work on this style of system, such as fine-tuning.

2 Related Work

This work sits at the intersection of CRSs, real-time automated assessment, KC generation, and LLMs. All of these domains individually encompass broad swaths of work but the study of their intersection is relatively limited.

2.1 Classroom Response Systems

CRSs have been deployed in various forms at various grade levels in classrooms around the world to improve learning outcomes and engagement. CRSs can take many forms, from simple multiple-choice “clicker”-style CRSs to more elaborate systems that support freeform text answers (Sianturi and Hung, 2023; Tyagi and Alyammahi, 2025; Serrada-Sotil et al., 2025). These CRSs can improve student outcomes in these lecture environments in a number of ways. They can improve engagement, cognitive outcomes such as learning gains, and non-cognitive outcomes such as creating comfort in responding to classroom exercises (Serrada-Sotil et al., 2025; England et al., 2017). However, many of these

An instructor posed the following question during an undergraduate computer science lecture:
 {question}
 Create a list of knowledge components covered by an ideal response to this question. Your answer should consist of ONLY the list of knowledge components described in a few words each.

Figure 2: The KC generation prompt for the No RAG condition.

An instructor posed the following question during an undergraduate computer science lecture:
 {question}
 The following context was given for the question:
 {RAGcontent}
 Create a list of knowledge components covered by an ideal response to this question. Only generate knowledge components that you think are explicitly covered by this question and the provided context. Your answer should consist of ONLY the list of knowledge components described in a few words each.

Figure 3: The KC generation prompt for the RAG condition.

existing classroom response systems that are deployed during lectures are limited to formats such as multiple-choice questions that may not encourage students to think deeply about course material or may only support relatively shallow analyses of text responses that are of limited use for measuring student knowledge. A system that could solicit and analyze short self-explanation responses from students could greatly improve interactivity, measure learning in lectures, provide immediate feedback to students, and allow instructors to perform real-time instructional pivots during lectures based upon student understanding of course material. Self-explanation has been shown to be a powerful tool for improving student learning over other question formats (Chi et al., 1994; Chi and Wylie, 2014), but presents more difficulty in analysis.

2.2 Assessing Short Self-explanations with Automated Short Answer Assessment

Assessing short self-explanations in real time for such a classroom response system presents a substantial technological hurdle for several reasons. Much of the existing work on assessing student written responses investigates the effectiveness of automated assessment in homework, exams, longer essay questions, or intelligent tutoring systems intended to be used during testing or outside of the classroom (Gao et al., 2023; Urrutia et al., 2025; Chamieh et al., 2024; Bexte et al., 2024). In com-

parison, student explanations posed in the classroom tend to present some unique challenges. Student self-explanations in this format tend to be very short, fragmented, may use implication that can obscure semantic meaning in order to answer questions more quickly, and contain frequent errors made in the haste of responding (Esiason et al., 2026a). Assessing many responses like this in a lecture format with many students may simply not be feasible for an instructor to do themselves due to time constraints. Additionally, probabilistic assessments have some drawbacks. If we wish to automate the process of assessing this type of response during a lecture, then we must do so with strong theoretical grounding such that we minimize errors that risk misleading students or harming their confidence in themselves or the assessment tool (Kerslake et al., 2025; Chamieh et al., 2024; Stowe et al., 2024).

2.3 Knowledge Components and Generation

Knowledge components represent a promising avenue for automatically assessing student responses for the presence of key units of knowledge or understanding. KCs have been used across a number of disciplines in the past to assess responses to a number of different question formats such as multiple-choice, short answer, or coding exercises (Koedinger et al., 2012). While sets of KCs have historically been laboriously generated by experts, recent work has demonstrated the effectiveness

Model	Scope	Sufficiency
ChatGPT-4o-mini (N)	0.881	0.778
ChatGPT-4o-mini (R)	0.637	0.941
Mistral-3B-Instruct (N)	0.778	0.593
Mistral-3B-Instruct (R)	0.709	0.754

Table 1: Median scope and sufficiency scores across all five generations, by model and RAG (R) or No RAG (N) condition. Scope and sufficiency range from [0,1], with values closer to 0 being desirable for Scope and values closer to 1 desirable for Sufficiency.

of KCs for arbitrary questions using LLMs (Esiason et al., 2026a; Moon et al., 2025; Moore et al., 2024a; Wei et al., 2025). This enables an automated path to producing CRS that can automatically generate KCs for instructor-generated questions. In addition to being adaptable to a variety of input and scalable, a KC tagging approach enables transparency of assessment since the KC tags can be traced and corrected throughout the assessment process.

2.4 Large Language Models

LLMs have risen to prominence in generating knowledge components, as they have in many other text-generation tasks. LLMs have proven to be reasonably effective in this task in various contexts and question formats (Esiason et al., 2026a; Moon et al., 2025; Moore et al., 2024a; Wei et al., 2025). Models used for this purpose must be carefully selected, however. While “black box” LLMs frequently provide state-of-the-art performance on a variety of tasks, open models are rapidly approaching parity (Kaggle, 2025) and there are concerns about using black-box models for educational purposes. Some of these concerns include security issues, such as data storage compliance or attack vulnerabilities (Li et al., 2024). Other concerns include hardware and support limitations. Larger models can avoid hardware costs by being deployed remotely in the cloud, but may come with API costs and are infeasible to run locally in the event of data security concerns. On the other hand, smaller models deployed locally may incur up-front hardware costs but ensure total control over student data. As a result, exploring both large and small or open- and closed-weight models is necessary.

There are many options to choose from, but two models have been selected for this work for the sake of expedience in human assessment. OpenAI’s ChatGPT-4o-mini (OpenAI, 2023, 2025) was se-

lected as the closed-weight model for use in this experiment due to its prior use in KC generation and short assessment work (Moon et al., 2025; Moore et al., 2024b). MistralAI’s Ministral-3-3B-Instruct (Liu et al., 2026) was chosen as the open-weight model for use in this experiment due to its reasonable performance in benchmarks (Kaggle, 2025) and open-weight status.

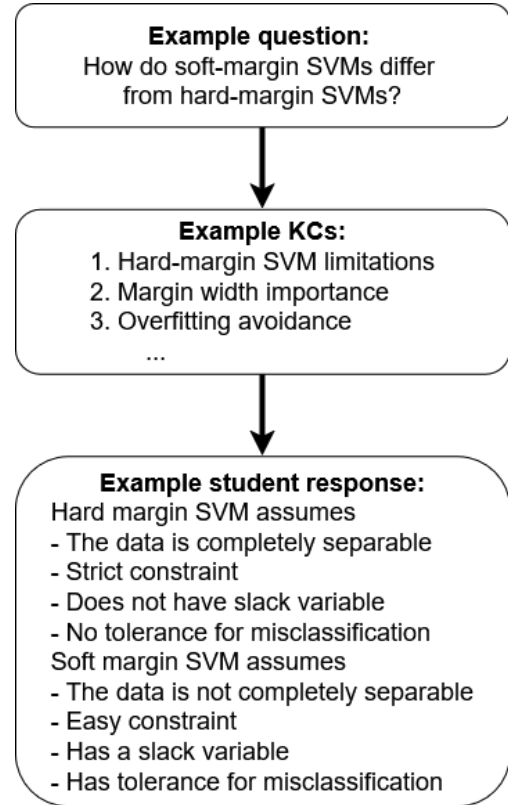


Figure 4: Example question, KCs generated from the question, and a related student self-explanation.

2.5 The EXPLAINIT Classroom Response System

EXPLAINIT is a multi-stage CRS that allows instructors to prepare short open-response questions, present them to students during lectures, gather responses from students, and provide automated feedback on those responses (Figure 1) (Carpenter et al., 2024). Instructors utilize the Instructor Dashboard to prepare questions ahead of time, and then present them to students who respond on a laptop or hand-held device of their choice through EXPLAINIT’s web platform (Figure 4). This engages students in self-explanation, allowing for deeper learning than pure lecturing or other CRSs that may use question formats that do not promote deeper learning, such as multiple-choice. LLM-driven,

	GPT (R)	Mistral (N)	Mistral (R)
GPT (N)	18.62, $p < 0.001$	4.45, $p = 0.035$	9.88, $p = 0.002$
GPT (R)	-	5.89, $p = 0.015$	1.31, $p = 0.253$
Mistral (N)	-	-	1.21, $p = 0.272$

Table 2: McNemar’s pairwise tests for differences in proportion of KCs that were out of scope by model and RAG (R) or No RAG (N) condition.

	GPT (R)	Mistral (N)	Mistral (R)
GPT (N)	12.25, $p < 0.001$	14.05, $p < 0.001$	0.08, $p = 0.775$
GPT (R)	-	39.93, $p < 0.001$	17.36, $p < 0.001$
Mistral (N)	-	-	5.97, $p = 0.015$

Table 3: McNemar’s pairwise tests for differences in proportion of KCs that were sufficient for identifying correct responses by model and RAG (R) or No RAG (N) condition.

immediate, personalized feedback then aids students in troubleshooting their response. Instructors are also presented with a real-time analytical dashboard of student responses, ensuring that a human is kept in the loop of the assessment process. An important design principle of EXPLAINIT is to create a low-overhead tool for instructors to use such that integration of the tool provides benefits without adding to their workload. This has informed the automated design choices underlying this system.

Recent work with this system has assessed its ability to provide real-time instructional support in undergraduate lectures (Esiason et al., 2026a,b) (Figure 1). In this system, instructors first utilize an authoring tool to prepare questions to pose to the class, categorized by topic and stored using EXPLAINIT web platform. Then, the system uses an LLM to derive a set of relevant KCs from the question. Next, the instructor uses the Instructor Dashboard to deliver a question to students during their lecture. Students respond to this question using a web interface on their phones, laptops, or other devices. At this point, the pipeline again uses an LLM to tag student responses with the boolean presence or absence of each KC from the set generated earlier. Once all student responses are assessed, an LLM is used to generate a summary of student performance based upon the question, the student responses, and their respective KC taggings. This KC approach allows for LLM assessments to be traced from question creation to summarization, and inherently supports human-in-the-loop interfacing by allowing instructors to drill down into student responses from the summaries, all the way back to the original question posed.

Prior work has found that the system performs

with a reasonable degree of accuracy with regards to agreement with human grader expectations for assessments of student responses and instructor preferences in how class-level summaries of student performance on questions should be formatted for easy and meaningful interpretation (Esiason et al., 2026a,b). While this previous work has established broad feasibility of this system, the reliability of the KC generation phase over repeated generation has not been established. The current work focuses exclusively on this stage.

3 Methods

All data collection activities were approved by IRB ANONYMIZED and were carried out in accordance with the relevant guidelines and regulations. Students were asked to review the study information sheet and electronically sign the IRB approved consent form to allow their data to be used for research purposes. Then all students were given a generic ID to interact with the system such that no personally identifiable information was included or collected by any systems external to the institution at which the research was being conducted. EXPLAINIT was deployed in one undergraduate computer science course on applied learning and data analytics in the fall semester of 2024. 27 instructor-generated questions were collected for use in this study through the Instructor Dashboard.

Models were selected to represent a balance between the state of the art in LLMs, prior work, and (in the case of Mistral-3-3B-Instruct) a relatively lightweight, open source option that could conceivably be deployed locally by an education agency. Both models were used as is out of the box. ChatGPT-4o-mini (OpenAI, 2025, 2023) was se-

Model	Median [Min,Max] KCs		
	Over (C>1)	Exact (C=1)	Under (C<1)
ChatGPT-4o-mini (N)	1 [0,4]	3 [0,5]	8 [0,16]
ChatGPT-4o-mini (R)	1 [0,3]	2 [0,6]	9 [1,13]
Mistral-3-3B-Instruct (N)	3 [0,10]	1 [0,7]	9 [3,24]
Mistral-3-3B-Instruct (R)	1 [0,4]	3 [0,7]	9 [3,25]

Table 4: Distributions of median over- and under-generated KCs per question as measured by consistency across 5 runs (C), by model and RAG (R) or No RAG (N).

Model	N Semantically Distinct KCs Generated	Median Semantically Distinct KCs per Question (Median [Min,Max])
ChatGPT-4o-mini (N)	347	12 [6,19]
ChatGPT-4o-mini (R)	325	13 [6,16]
Mistral-3-3B-Instruct (N)	374	12 [5,32]
Mistral-3-3B-Instruct (R)	395	13 [6,29]

Table 5: Total, median, minimum, and maximum number of KCs generated by model and RAG (R) or no RAG (N).

lected as a baseline based upon prior work with this system (Esiason et al., 2026a,b; Carpenter et al., 2024) and related assessment and KC generation tasks (). MistralAI’s Mistral-3-3B-Instruct (Liu et al., 2026) was chosen in order to evaluate the performance of an open source, relatively lightweight model that could be reasonably be expected to run locally on institution hardware rather than relying on third-party closed-source APIs. All models were run at a low temperature (0.1) in order to align with previous work (Esiason et al., 2026a) and increase the reliability of the system. Outputs of each model were limited to 2,000 tokens, although outputs were frequently shorter.

Each model was used on the same question sets to produce knowledge components by RAG and non-RAG approaches. This was done to compare the performance of the two approaches and attempt to control for previously identified issues with this pipeline generating sets of knowledge components that were out of scope of the question being asked (Esiason et al., 2026a). We theorized that a RAG approach would narrow the generated knowledge component sets to what could be considered fairly expected of students, given that these questions are deployed after the students have only just been introduced to the material either by assigned reading or the lecture itself.

For the RAG models, one reference material per question was obtained from one of three sources. All materials produced were in text or table format, with figures excluded. Firstly, we attempted to obtain reference material from the course textbook

(Tan et al., 2019) and transcribed the relevant paragraph(s) and/or tables. Occasionally, the questions referenced content that was not available in the textbook; in these cases reference materials were sourced from other comparable locations appropriate for an undergraduate student lecture (Simon et al., 2018; Boehmke). We attempted to retrieve comparably useful and relevant text for RAG across questions, but RAG material varied in length from a few sentences to several pages depending on the question.

Once RAG material were selected, the instructor questions were paired with their RAG material and ingested by the EXPLAINIT pipeline. The same prompt was used for each model, with the only variation being in the inclusion of the RAG material between RAG and No RAG conditions. In the No RAG condition, only the instructor question was passed to the LLM through the prompt (Figure 2). In the RAG condition, both the instructor question and the RAG material were passed to the LLM through the prompt (Figure 3). All models and RAG combinations were run 5 times for each input data, once with RAG utilizing course materials, and once without RAG. This resulted in 135 generated KC sets per model/RAG combination.

Two raters divided up the task of evaluating these KC sets along the metrics of scope and sufficiency. “Scope” intended to measure the whether or not a given KC was within the intent of the question being posed. For example, if the question was “what are the steps of k -fold cross validation” and one of the generated KCs involved comparing k -fold

	GPT (R)	Ministral (N)	Ministral (R)
GPT (N)	$V = 132.00, p = 0.577$	$V = 156.00, p = 0.583$	$V = 164.00, p = 0.562$
GPT (R)	-	$V = 103.50, p = 0.746$	$V = 139.00, p = 0.239$
Ministral (N)	-	-	$V = 133.50, p = 0.186$

Table 6: Wilcoxon signed-rank pairwise tests for differences in consistency of KC generation between models by RAG (R) or No RAG (N).

cross validation to other training algorithms then this KC would be out of scope. Sufficiency, on the other hand, is intended to measure whether or not a given set of KCs is sufficient for identifying a correct response from a student (regardless of whether or not it contains KCs that are out-of-scope). For example, if the question is “what are the steps of k -fold cross validation” and the LLM does not generate a KC corresponding to “divide the training data into k folds” then the generated KC set would be insufficient to identify a correct response since one of the first steps of k -fold cross validation is to divide the training data into k folds. Inter-rater reliability was computed using Gwet’s AC1 (Gwet, 2002) using a stratified sample of the KCs such that each question and a respective KC set were each represented. Questions were also grouped by one expert grader using Bloom’s revised taxonomy (Anderson and Krathwohl, 2001) as a guide to creating more narrowly-defined question types. Questions were grouped into the categories of procedural knowledge (e.g. recalling the steps of an algorithm), comparison (e.g. comparing two algorithms), explaining specific concepts of an algorithm (e.g. modulating the number of starting clusters in a clustering algorithm), or factual recall (e.g. recalling the definition of a term). While Bloom’s revised taxonomy contains other criteria for categorizing questions, these four categories were the only ones observed in the data.

All knowledge components generated were qualitatively semantically matched to each other by an expert grader where appropriate in order to evaluate consistency of KC generation. For example, if two or more runs generated closely-related variations of the knowledge component “divide data into k folds” for the question “what are the steps of k -fold cross validation?”, then these knowledge components were matched. Consistency was measured as the number of times a matched KC was generated divided by the number of runs. Ideally this value would approach 1, indicating that a KC was likely generated once per run. A value over 1

would indicate overly repetitive generation, while a value under 1 indicates inconsistent generation.

4 Results

One generated KC set was excluded from analysis of the Ministral-3-3B-Instruct model with RAG due to substantial generation failure. An exorbitant number of repetitive KCs were generated in this case (126 in total), indicating that the model may have gotten stuck in a generation loop due to the low temperature. No other models or runs demonstrated any otherwise equivalently catastrophic looping failures, although one run of one question (“what are the common properties of similarity?”) for the Mistral-3-3B-Instruct produced completely topically irrelevant content in the form of software engineering advice.

IRR was achieved in the first round of coding for sufficiency (AC1=0.777). Scope required an additional round of coding on a second similarly stratified sample, at which point sufficient IRR was achieved (AC1=0.704). Once sufficient IRR was achieved for both metrics, the remaining data were divided up among the raters and coded. Differences in scope and sufficiency shown in Table 1 were measured as statistically significant by Cochran’s Q test ($Q = 22.654, p < 0.001$ and $Q = 47.571, p < 0.001$, respectively). Post-hoc McNemar tests (Tables 2 and 3) demonstrate pairwise differences between models and conditions.

All models produced similar numbers of semantically distinct KCs across all runs, with the Ministral-3-3B-Instruct model producing slightly more KCs (374 without RAG, 395 with RAG) than GPT-4o-mini (347 without RAG, 325 with RAG). Both models without RAG produced a median of 12 KCs per question while both models with RAG produced a median of 13 KCs per question, although the range in number of generated KCs differed (Table 5). All models had the same overall median consistency score of 0.60. Consistency of generation for GPT-4o-mini without RAG, GPT-4o-mini with RAG, Ministral-3-3B-Instruct

without RAG, and Ministral-3-3B-Instruct with RAG ranged from [0.2,2.8], [0.2,2.4], [0.2,5.0], and [0.2,3.0], respectively. The questions that produced the most consistent KC sets (median consistency ≥ 0.80) varied by model with no readily apparent pattern. Pair-wise Wilcoxon signed rank tests were performed between model and RAG conditions (Table 6) but revealed no statistically significant difference in consistency of generation. While the dataset contains several types of questions (such as explaining the steps of an algorithm, explaining how modulating a parameter will impact algorithm performance, etc.) there appeared to be no statistical difference in the distribution of types of questions (comparison, algorithm component explanation, algorithm steps, and factual recall) between the most and least consistent questions (Fisher’s Exact $p = 1.00$).

5 Discussion

With regard to RQ1 (“Does including course materials as RAG context for these models result in the generation of higher quality KCs as measured by the scope of the generated content and the sufficiency of the generated KCs in capturing correct responses?”), clear patterns can be observed in RAG vs. No RAG as well as model choice conditions. Statistical testing indicates that the inclusion of RAG content can substantially limit the generation of out of scope KCs. In GPT-4o-mini’s case, out of scope KCs were reduced by 24.4%. In Ministral-3-3B-Instruct’s case, out of scope KCs were reduced by 6.9%. GPT-4o-mini statistically outperforms both conditions of the Ministral model, although the difference is fairly small between the RAG conditions (0.072). The inclusion of RAG material similarly improves the sufficiency of questions in capturing correct responses, with the RAG conditions of GPT-4o-mini and Ministral showing no statistically significant difference in performance. These findings indicate that, for these metrics, GPT-4o-mini and Ministral are near or equivalent in performance and thus both potentially suitable for further exploration of this task. As a result, it may be feasible to allow institutions of higher education to maintain control over their own data using smaller, open-weight models while using a CRS like EXPLAINIT. Additionally, the findings indicate that the inclusion of course material excerpts as RAG content improve the generated KCs over the No RAG condition. Since course material would pre-

sumably be readily available to these models as it has been in this experiment, RAG presents an accessible path to improving KC generation consistency.

With regard to RQ2 (“How consistent are the output of these models over multiple generation cycles?”), consistency was moderate and there was no statistically significant difference in consistency across models and conditions. This may indicate that while issues regarding scope and sufficiency can be reduced with RAG material, issues of consistency even at very low temperatures may require other avenues of model improvement in addition to RAG such as fine-tuning, different prompting strategies such as zero-shot with instructor examples, or purpose-built models. This work may also have identified a relationship between proximity of the KC to the start of the generated KC set and consistency, although this was not explicitly tested or controlled for in this work. We observed that the most consistently generated KCs appear to generally be placed earlier in the lists of KCs produced than those of lower consistency. Because of the sequential nature of the generation of lists of KCs by LLMs in this study, we theorize that the apparent high stability of KCs that are placed first in the list may be due to these KC tokens having high probabilities given the previously-generated text and low temperature. While no statistical tests were performed on this topic, consistency scores were observed to largely decrease as the KCs moved further from the start of the KC set. This may indicate that KC sets are likely to start with high-probability tokens before steadily degenerating to lower-probability, more inconsistently generated KCs as inference proceeds. This could present an avenue for improving consistency of output by eliminating low-probability tokens at time of generation, such as by examining token log probabilities during generation.

With regard to the final RQ (“What, if any, other qualitative or quantitative patterns appear in generated KCs with regards to their quality?”), no visually obvious quantitative or qualitative patterns appeared in the generation of KCs vs. the types of questions analyzed. This may indicate that the models tested are equally as effective in generating KCs for the given dataset regardless of whether the question is oriented towards formats such as factual recitation, exploring the impact of altering or explaining individual algorithmic components, asking students to reason through the steps of an al-

gorithm, or offering a comparative analysis of two or more algorithms or methods. Such flexibility will be useful, as this would theoretically allow instructors to pose a wide range of types of questions based upon the material being delivered.

6 Limitations

Several limitations apply to this work. Firstly, the question set is varied in computer science topic and question type (factual recall, explanation of algorithm components, etc.) but still relatively small with only 27 questions. This limits statistical power when attempting to examine relationships between metrics and question type. Secondly, one catastrophic KC set generation and one completely off topic generation were observed in the open-weight model. While this may simply indicate the existence of relatively rare generation errors, these errors cannot be overlooked in an educational setting and warrant further investigation into eliminating these catastrophic generations or enhancing the human-in-the-loop aspects of the EXPLAINIT system. Additionally, RAG materials varied in quality. The size of the passages varied substantially from a few sentences or bullet points to several pages and could contain irrelevant content that could misguide the LLMs generation process, resulting in KCs that are sourced from the RAG materials but ultimately out of scope of the spirit of the question being asked. Furthermore, KCs were taken as given based upon prior work and were not validated on student data. While the expert graders evaluating the data are knowledgeable about the subject matter of the questions, a more thorough analysis of the KCs on student data would improve the robustness of the status of the generated KCs as “true” KCs. Lastly, this work failed to show within-model boosts to consistency through the use of RAG so further exploration of methods to improve consistency must be pursued. But this work highlighted a potential avenue for improving this issue by identifying a potential pattern in KC generation that could be studied and exploited in order to limit generations to KCs that the LLM is more “confident” in by restricting KCs to those with high token log probabilities. Further exploration would also inform a deeper understanding of consistency of generation, which could in turn inform how a practitioner would interpret or even tweak the KCs produced by this CRS.

7 Conclusion and Future Work

The inclusion of a pipeline allowing instructors to deploy and automatically assess self-explanation questions in CRSs has significant potential to improve learning outcomes over traditional, less interactive lecture experiences or CRSs with less engaging question types by increasing interactivity and prompting students to explain their understanding of course topics and materials in their own words. Assessing the responses to these questions accurately and consistently in real time during lectures poses a substantial technological challenge, but LLMs supported by RAG using course materials may present a promising path towards tackling this challenge through the flexible automated generation of knowledge components.

Our work addresses this novel problem by evaluating the quality of KCs generated by two popular LLMs with or without relevant RAG material drawn from course materials along several metrics: whether or not the generated KCs are within the scope of the question, whether or not the generated KCs are sufficient to identify a correct answer, and the consistency of the generation of these KCs across multiple rounds of generation. Results indicate that the LLM-generated KCs are of higher quality when relevant course material is provided for RAG. Additionally, while the closed-weight model’s performance is statistically significantly better than the open-weight comparison model the gap is small. Furthermore, no statistically significant difference in consistency of KC generation is observed between RAG conditions. These results indicate that both large closed-weight and smaller open-weight models show promise in this task and that fine-tuning or improvements to the quality of RAG material may be necessary to improve consistency and quality further.

Acknowledgments

This research was supported by funding from the National Science Foundation under grants DUE-2111473, DUE-2111216, and CISE Graduate Fellowship CISE-2313998. 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. Additionally, the authors would like to thank Rebecca Zarch and Tyler Clark of SageFox Consulting Group for their contributions in ideation.

References

- Lorin W. Anderson and David R. Krathwohl. 2001. *A Taxonomy for Learning, Teaching, and Assessing : a Revision of Bloom's Taxonomy of Educational Objectives*. Longman.
- Marie Bexte, Andrea Horbach, Lena Schützler, Oliver Christ, and Torsten Zesch. 2024. [Scoring with confidence? – exploring high-confidence scoring for saving manual grading effort](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 119–124, Mexico City, Mexico. Association for Computational Linguistics.
- Bradley Boehmke. [Regularized regression](#). Accessed: 2026.
- Dan Carpenter, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng, and James Lester. 2024. [Assessing student explanations with large language models using fine-tuning and few-shot learning](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 403–413, Mexico City, Mexico. Association for Computational Linguistics.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- Michelene T. H. Chi and Ruth Wylie. 2014. [The icap framework: Linking cognitive engagement to active learning outcomes](#). *Educational Psychologist*, 49(4):219–243.
- Michelene T.H. Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian Lavancher. 1994. [Eliciting self-explanations improves understanding](#). *Cognitive Science*, 18(3):439–477.
- Benjamin J. England, Jennifer R. Brigati, and Elisabeth E. Schussler. 2017. [Student anxiety in introductory biology classrooms: Perceptions about active learning and persistence in the major](#). *PLOS One*, 12(8).
- Jordan Esiason, Priyanka Khare, Claire Aguiar, Dan Carpenter, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng, and James Lester. 2026a. [An explanation-based classroom response system for real-time analysis of undergraduate students' natural language explanations](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(48):40822–40830.
- Jordan Esiason, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng, Yeil Jeong, and James Lester. 2026b. [Topic-level feedback summarization for an explanation-based classroom response system](#). In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.2, SIGCSE TS 2026*, page 1325–1326, New York, NY, USA. Association for Computing Machinery.
- Rujun Gao, Hillary E. Merzdorf, Saira Anwar, M. Cynthia Hipwell, and Arun Srinivasa. 2023. [Automatic assessment of text-based responses in post-secondary education: A systematic review](#). *ArXiv*, abs/2308.16151.
- Scott Grissom, Renée Mccauley, and Laurie Murphy. 2017. [How student centered is the computer science classroom? a survey of college faculty](#). *ACM Trans. Comput. Educ.*, 18(1).
- Kilem L. Gwet. 2002. [Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity](#).
- Nathaniel J. Hunsu, Olusola Adesope, and Dan James Bayly. 2016. [A meta-analysis of the effects of audience response systems \(clicker-based technologies\) on cognition and affect](#). *Computers Education*, 94:102–119.
- Kaggle. 2025. [Mmlu open benchmarks](#). Accessed: 2025-03-12.
- Chris Kerslake, Paul Denny, David H. Smith, Juho Leinonen, Stephen MacNeil, Andrew Luxton-Reilly, and Brett A. Becker. 2025. [Exploring student reactions to llm-generated feedback on explain in plain english problems](#). In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1, SIGCSETS 2025*, page 575–581, New York, NY, USA. Association for Computing Machinery.
- Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. 2012. [The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning](#). *Cognitive Science*, 36(5):757–798.
- Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. [Llm-pbe: Assessing data privacy in large language models](#). *Proc. VLDB Endow.*, 17(11):3201–3214.
- Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. [Ministral 3](#). *Preprint*, arXiv:2601.08584.
- Hyeongdon Moon, Richard Lee Davis, Seyed Parsa Neshaei, and Pierre Dillenbourg. 2025. [Using large multimodal models to extract knowledge components for knowledge tracing from multimedia question information](#). In *Proceedings of the 18th International Conference on Educational Data Mining*, pages 342–353, Palermo, Italy. International Educational Data Mining Society.

- Steven Moore, Robin Schmucker, Tom Mitchell, and John Stamper. 2024a. [Automated generation and tagging of knowledge components from multiple-choice questions](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale, L@S '24*, page 122–133, New York, NY, USA. Association for Computing Machinery.
- Steven Moore, Robin Schmucker, Tom Mitchell, and John Stamper. 2024b. [Automated generation and tagging of knowledge components from multiple-choice questions](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale, L@S '24*, page 122–133, New York, NY, USA. Association for Computing Machinery.
- OpenAI. 2023. Gpt-4 technical report. <https://cdn.openai.com/papers/gpt-4.pdf>. Accessed: 2025-04-23.
- OpenAI. 2025. Chatgpt-4o-mini api. <https://platform.openai.com/docs>. Accessed: 2025-04-23.
- Jaime Serrada-Sotil, Juan Antonio Huertas Martínez, and Miriam Granado-Peinado. 2025. [Do audience response systems truly enhance learning and motivation in higher education? a systematic review](#). *Humanities and Social Sciences Communications*, 12.
- Alex Dharmawan Sianturi and Ruei-Tang Hung. 2023. [The challenges of using kahoot! in teaching and learning in higher education – a systematic review](#). In *Proceedings of the 6th International Conference on Digital Technology in Education, ICDTE '22*, page 72–77, New York, NY, USA. Association for Computing Machinery.
- Laura Simon, Derek Young, and Iain Pardoe. 2018. [2.1 - what is simple linear regression?](#) Accessed: 2026.
- M. Stains, J. Harshman, M. K. Barker, S. V. Chasteen, R. Cole, S. E. DeChenne-Peters, M. K. Eagan, J. M. Esson, J. K. Knight, F. A. Laski, M. Levis-Fitzgerald, C. J. Lee, S. M. Lo, L. M. McDonnell, T. A. McKay, N. Michelotti, A. Musgrove, M. S. Palmer, K. M. Plank, and 12 others. 2018. [Anatomy of stem teaching in north american universities](#). *Science*, 359(6383):1468–1470.
- Kevin Stowe, Benny Longwill, Alyssa Francis, Tatsuya Aoyama, Debanjan Ghosh, and Swapna Somasundaran. 2024. [Identifying fairness issues in automatically generated testing content](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 232–250, Mexico City, Mexico. Association for Computational Linguistics.
- Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2019. *Introduction to Data Mining, Second Edition*. Pearson Education, Inc.
- Sanjay Kumar Tyagi and Aishah Mohamed Saeed Abdulla Alyammahi. 2025. [Adoption of interactive learning tools in higher education: Insights for an ai-driven future](#). *2025 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pages 681–686.
- Felipe Urrutia, Cristian Buc, Roberto Araya, and Valentin Barriere. 2025. [Unsupervised automatic short answer grading and essay scoring: A weakly supervised explainable approach](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 38–54, Vienna, Austria. Association for Computational Linguistics.
- Yumou Wei, Paulo Carvalho, and John Stamper. 2025. [Kcluster: An llm-based clustering approach to knowledge component discovery](#). In *Proceedings of the 18th International Conference on Educational Data Mining*, pages 228–240, Palermo, Italy. International Educational Data Mining Society.

Estimating LLM Grading Ability and Response Difficulty in Automatic Short Answer Grading via Item Response Theory

Longwei Cong¹, Sonja Hahn¹, Sebastian Gombert¹, Leon Camus¹,
Hendrik Drachsler^{1,2}, Ulf Kroehne^{1,3}

¹DIPF | Leibniz Institute for Research and Information in Education

²Faculty of Computer Science, Goethe University Frankfurt

³Chemnitz University of Technology

{l.cong,s.hahn,s.gombert,l.camus,h.drachsler,u.kroehne}@dipf.de

Abstract

Automated short answer grading (ASAG) with large language models (LLMs) is commonly evaluated with aggregate metrics such as macro-F1 and Cohen’s kappa. However, these metrics provide limited insight into how grading performance varies across student responses of differing grading difficulty. We introduce an evaluation framework for LLM-based ASAG based on item response theory (IRT), which models grading correctness as a function of latent grader ability and response grading difficulty. This formulation enables response-level analysis of where LLM graders succeed or fail and reveals robustness differences that are not visible from aggregate scores alone. We apply the framework to 17 open-weight LLMs on the SciEntsBank and Beetle benchmarks. The results show that even models with similar overall performance differ substantially in how sharply their grading accuracy declines as response difficulty increases. In addition, confusion patterns show that errors on difficult responses concentrate disproportionately on the *partially_correct_incomplete* label, indicating a tendency toward intermediate-label collapse under ambiguity. To characterize difficult responses, we further analyze semantic and linguistic correlates of estimated difficulty. Across both datasets, higher difficulty is associated with weaker semantic alignment to the reference answer, stronger contradiction signals, and greater semantic isolation in embedding space. Overall, these results show that item response theory offers a useful framework for evaluating LLM-based ASAG beyond aggregate performance measures.

1 Introduction

Automatic short answer grading (ASAG) is the task of using computational methods to assess short free-text student responses to content questions (Burrows et al., 2015). Earlier ASAG approaches relied on handcrafted features and classical machine learning models (Sultan et al., 2016;

Mohler et al., 2011), followed by neural architectures and encoder-based pre-trained language models (PLMs) such as BERT (Gombert et al., 2023; Camus and Filighera, 2020). More recently, generative large language models (LLMs) have attracted increasing attention for ASAG because of their strong language understanding and multi-step reasoning capabilities (Cong et al., 2026b; Frohn et al., 2025; Ferreira Mello et al., 2025).

Despite this progress, evaluation in ASAG remains dominated by aggregate metrics such as accuracy, macro-F1, and agreement coefficients (Bonthu et al., 2021). While useful for summarizing overall system performance, these measures provide only a coarse view of grading behavior and reveal little about how performance varies across responses of differing grading difficulty (Rodriguez et al., 2021; Choi et al., 2026). This limitation is particularly important in ASAG, where student answers are often brief, diverse, and semantically ambiguous (Burrows et al., 2015). Consequently, two models with similar overall scores may nevertheless differ substantially in how well they handle genuinely difficult responses.

This motivates a more measurement-oriented perspective on LLM-based ASAG. In particular, a useful evaluation framework should help characterize which student responses are difficult to grade and whether LLM graders with similar average performance are equally robust across the difficulty spectrum. Existing ASAG research has paid relatively little attention to these questions at the response level (Haller et al., 2022).

To address this gap, we draw on item response theory (IRT), a psychometric framework that models observed outcomes as a function of latent ability and item difficulty. Prior work has shown that IRT can support more informative evaluation in NLP when benchmark instances vary substantially in difficulty (Lalor et al., 2016, 2018). We argue that IRT is well suited to ASAG because it separates latent

grader ability from response difficulty, enabling a more fine-grained analysis of grading behavior than aggregate metrics alone.

In this work, we apply an IRT-based framework to LLM-based ASAG to study grading performance at the level of individual student responses. Specifically, we make the following contributions:

- We introduce an IRT-based perspective for evaluating LLM-based ASAG, estimating latent grader abilities and response grading difficulties across benchmark datasets and multiple LLM graders.
- We analyze how grading performance changes across response difficulty levels and show that LLM graders differ substantially in their robustness as difficulty increases.
- We examine semantic and linguistic correlates of response difficulty and find that difficult responses are consistently associated with weaker semantic alignment to the reference, stronger contradiction-related signals, and greater embedding-space isolation.
- We provide error analyses showing that, on high-difficulty responses, misclassifications are not randomly distributed but increasingly concentrate in the `partially_correct_incomplete` category.

2 Background

2.1 Evaluation Methods in LLM-based ASAG

Evaluation in ASAG has traditionally focused on benchmark performance over labeled response sets, typically reported using aggregate metrics such as accuracy, F1, Cohen’s kappa, RMSE, and correlation-based measures (Haller et al., 2022; Burrows et al., 2015; Sultan et al., 2016). Recent LLM-based ASAG studies largely follow this paradigm, assessing LLM graders through overall label agreement or classification performance (Ferreira Mello et al., 2025; Frohn et al., 2025; Cong et al., 2026b).

Yet recent research on LLM evaluation suggests that aggregate benchmark scores alone may provide only a limited view of model capability (Zhou et al., 2025; Li et al., 2025). Prior work shows that benchmark-level averages can obscure substantial variation in item properties such as difficulty and discriminability, while offering limited separability

among strong models (Lalor et al., 2016, 2018). Similarly, recent work on LLM-as-a-Judge argues that LLM-based evaluators should themselves be evaluated systematically, with reliability, bias, and robustness as central concerns (Gu et al., 2025).

Within ASAG, however, the response-level determinants of grading difficulty remain underexplored, particularly for LLM-based grading. Understanding which properties of student answers make them difficult to grade is important both for interpretability and for deployment, especially in human-in-the-loop settings where difficulty estimates could guide targeted review and improve grading efficiency. Although recent work has examined confidence and uncertainty in ASAG (Funayama et al., 2022; Bexte et al., 2024; Cong et al., 2026a), these signals are typically model-derived and thus only indirectly informative about the characteristics of the response itself. Analyzing difficulty as a property of the student response instead offers a more interpretable and educationally grounded perspective on the failure modes of LLM-based grading.

2.2 Item Response Theory

Item Response Theory (IRT) provides a principled framework by disentangling latent responder ability from item properties (van der Linden, 2016). In the simplest Rasch formulation, the probability that responder m correctly answers item n is modeled as

$$P(y_{mn} = 1 \mid \theta_m, b_n) = \sigma(\theta_m - b_n),$$

where θ_m denotes the latent ability of responder m , b_n the difficulty of item n , and $\sigma(\cdot)$ the logistic sigmoid. More flexible variants, such as 2PL models, additionally model item discrimination, allowing some items to be more informative than others (Hambleton and Swaminathan, 2013).

Recent work has begun to apply IRT to LLM evaluation in several ways, including diagnosing the reliability of LLM-as-a-judge systems (Choi et al., 2026) and modeling latent LLM abilities together with query difficulty and discrimination for routing and performance prediction (Chen et al., 2025). In educational technology, related work has also shown that IRT can be integrated with LLM-based modeling to predict or align item difficulty in assessment settings (Scarlato et al., 2025). Collectively, these developments suggest that psychometric modeling offers a robust lens for analyzing LLM behavior through a measurement-grounded perspective.

3 Method

3.1 Datasets and LLMs

We evaluate our approach on the five-way SciEnts-Bank and Beetle datasets (Dzikovska et al., 2013), two widely used benchmarks for ASAG. Released as part of the SemEval student-response analysis task, both datasets contain science-domain questions, reference answers, and student responses annotated by experts. An important property of these benchmarks is their fine-grained five-way label space, which enables analysis beyond binary correctness. Each instance consists of a question, a single reference answer, a student response, and a gold label from the set `correct`, `contradictory`, `partially_correct_incomplete`, `irrelevant`, and `non_domain`. An important difference between the two datasets is that Beetle is drawn from a relatively narrow tutorial domain in basic electricity and electronics, whereas SciEntsBank covers assessment responses across a substantially broader set of science topics (Dzikovska et al., 2013).

As LLM graders, we select 17 open-weight models covering multiple model families including Google, Qwen, Microsoft, Meta-Llama, MistralAI, and OpenChat, with parameter sizes ranging from 0.8B to 14B. The complete list of models is provided in Table 4 in Appendix A.2. The goal is to capture substantial diversity in grading behavior, including variation associated with model size, architectural lineage, and training characteristics.

3.2 IRT Estimation of Grader Ability and Response Difficulty

We treat each LLM as a grader and each student response as an item to be evaluated. Given a question, a reference answer, and a student response, each model is prompted to assign exactly one label from the five benchmark categories. Decoding is performed greedily with temperature set to 0 to ensure deterministic and reproducible model outputs.

Let $i \in \{1, \dots, M\}$ index LLM graders and $j \in \{1, \dots, J\}$ index student-response instances. For each grader–response pair, we define a binary observation

$$y_{ij} = \begin{cases} 1 & \text{if grader } i \text{ is correct on response } j, \\ 0 & \text{otherwise.} \end{cases}$$

Because multiple student responses are associated with the same question, we include grader-specific testlet effects to capture residual question-specific variation beyond overall grader ability and

response difficulty. Specifically, we extend the Rasch model with testlet effects and fit a testlet response theory (Wainer et al., 2007) variant:

$$P(y_{ij} = 1) = \sigma(\theta_i - b_j + u_{i,t(j)}),$$

where θ_i denotes the latent grading ability of grader i , b_j denotes the grading difficulty of student-response instance j , and $t(j)$ denotes the question-level testlet to which response instance j belongs. The term $u_{i,t(j)}$ is a grader-specific deviation associated with testlet $t(j)$, allowing graders to exhibit question-specific performance offsets beyond their overall ability. We use a 1PL testlet formulation rather than a 2PL variant because a 2PL model would introduce an additional discrimination parameter for each response instance. Given the large number of response instances, we therefore adopt the 1PL formulation and account for question-level dependence through testlet deviations.

Parameters are estimated by minimizing the regularized negative log-likelihood:

$$\begin{aligned} \mathcal{L} = & - \sum_{i,j} \left[y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij}) \right] \\ & + \frac{\lambda_\theta}{2} \|\theta\|_2^2 + \frac{\lambda_b}{2} \|b\|_2^2 + \frac{\lambda_u}{2} \|u\|_2^2, \end{aligned} \quad (1)$$

Following our implementation, we set $\lambda_\theta = \lambda_b = 1.0$ and $\lambda_u = 5.0$. To ensure identifiability, θ and b are mean-centered within the fitted dataset, and the testlet effects are centered across graders within each testlet. Optimization is performed using LBFGS (Liu and Nocedal, 1989).

3.2.1 Parameter Recovery

To assess whether the latent parameters are recoverable in practice, we conduct a parameter recovery study using the fitted IRT model as the data-generating process, following standard practice in IRT evaluation (Harwell, 1997). Specifically, we treat the fitted model as a generative data source and conduct a simulation-based parameter recovery analysis. For each replication, we use the estimated grader abilities, response difficulties, and grader-specific testlet effects to compute the probability of correct grading for every grader–response pair under the fitted 1PL testlet model. We then sample a new binary correctness matrix from these probabilities, re-fit the model to the simulated data using the same estimation procedure as for the original data, and compare the recovered parameters with

the original fitted parameters after alignment to a common scale.

Recovery is quantified using Pearson correlation, RMSE, and MAE, after aligning the recovered parameters to the original scale via mean and standard-deviation matching. We report these metrics for grader abilities and response difficulties. This analysis assesses whether the inferred latent quantities are sufficiently recoverable to support downstream interpretation.

3.2.2 Split-Half Stability

To assess empirical robustness on real data, we conduct a split-half stability analysis. Whereas parameter recovery evaluates recoverability under synthetic responses generated from the fitted model, split-half stability examines whether the inferred parameters remain consistent when the model is re-estimated on disjoint subsets of the observed data.

We use two complementary procedures. To assess grader-ability stability, we randomly split responses into two halves, fit the model separately on each half using the full set of graders, and compare the resulting ability estimates. Because our main model includes question-level testlet effects, response splits are performed within each question. To assess response-difficulty stability, we randomly split graders into two halves, fit the model separately on each subset using the full set of responses, and compare the resulting difficulty estimates.

We repeat both procedures over 10 random replications and quantify stability using Pearson and Spearman correlations, together with RMSE and MAE after mean–standard deviation alignment to account for scale indeterminacy. High split-half agreement indicates that the inferred latent quantities capture reproducible structure in the observed data.

3.3 Difficulty-Conditioned Analysis of LLM Grader Behavior

To analyze how grading behavior changes as response difficulty increases, we use the estimated response difficulty b_j as the primary stratification variable. We partition responses into quantile-based difficulty bins and compute each grader’s empirical accuracy within each bin. This stratified evaluation complements aggregate metrics by revealing not only whether performance declines with increasing difficulty, but also whether graders differ in their sensitivity to difficult responses.

Using these bins, we perform two analyses. First, we examine the relationship between difficulty and grading accuracy to assess how performance changes across the difficulty spectrum. Second, we inspect bin-specific confusion matrices to identify how error patterns shift with increasing difficulty.

3.4 Semantic and Linguistic Correlates of Difficulty

To examine which response properties are associated with grading difficulty, we extract a focused set of semantic and linguistic features for each student response, motivated by prior work in ASAG (Mohler and Mihalcea, 2009; Burrows et al., 2015; Dzikovska et al., 2013). These features include response length, lexical diversity, unigram and bigram overlap with the reference answer, semantic similarity to the reference answer, NLI-based features computed on the reference–response pair, embedding-space neighborhood features, and the number of missing reference segments. Sentence-embedding features are computed using all-MiniLM-L6-v2 (Reimers and Gurevych, 2019), and NLI-based features are obtained using a ModernBERT-based NLI model (Warner et al., 2025). A detailed description of the features is provided in Appendix A.1.

We treat these features as correlates of estimated response difficulty. For each dataset, we compute both Pearson and Spearman correlations between each feature and the estimated difficulty b_j , and apply the Benjamini–Hochberg procedure to control the false discovery rate under multiple comparisons (Benjamini and Hochberg, 1995).

4 Results

We report three sets of results. First, we assess the stability of the proposed IRT framework and its latent estimates of grader ability and response difficulty. Second, we examine how LLM grading performance degrades as difficulty increases, including changes in confusion patterns on more difficult responses. Third, we analyze how estimated difficulty relates to semantic alignment and response atypicality.

4.1 Parameter Recovery and Split-Half Stability

The fitted IRT model converged successfully on both datasets and produced a broad range of latent grader-ability estimates across the 17 evaluated LLMs, indicating substantial heterogeneity in

Parameter	Pearson r	RMSE	MAE	Conv.
<i>SciEntsBank</i>				
grader ability (θ)	0.999	0.029	0.024	1.000
response difficulty (b)	0.896	0.505	0.404	1.000
<i>Beetle</i>				
grader ability (θ)	0.998	0.036	0.029	1.000
response difficulty (b)	0.895	0.497	0.398	1.000

Table 1: Parameter recovery results over 10 replications on SciEntsBank and Beetle. Values are mean recovery statistics after mean–std alignment to the ground-truth scale. Conv. denotes convergence rate.

grading ability. The full model-level results are reported in Table 4 in Appendix A.2.

Parameter recovery results in Table 1 further support the stability of these latent estimates under the fitted model. For grader ability, recovery was nearly perfect on both datasets, with mean aligned Pearson correlations of 0.999 on SciEntsBank and 0.998 on Beetle, together with very low RMSE/MAE values (0.029/0.024 and 0.036/0.029, respectively). Response difficulty estimates were also recovered well, with Pearson correlations of 0.896 on SciEntsBank and 0.895 on Beetle, and comparable error levels (RMSE = 0.505 and 0.497; MAE = 0.404 and 0.398). All recovery runs converged successfully on both datasets.

Complementary split-half analyses on the observed data yielded a similar pattern (Table 2). Grader ability estimates were highly stable across response halves, whereas response difficulty estimates showed moderate-to-strong stability across grader halves, indicating that the inferred difficulty structure is reproducible but noisier than the model ability scale. Taken together, these results suggest that the IRT framework yields identifiable and empirically robust estimates of both latent grading ability and response difficulty, providing a sound measurement-oriented foundation for the subsequent analyses.

4.2 Differential Performance Degradation Across Difficulty Levels

LLM graders differ substantially in how their grading performance degrades as response difficulty increases. We grouped responses into five quantile-based bins according to their estimated difficulty b , from the easiest ($B1$) to the hardest ($B5$). Across both SciEntsBank and Beetle, accuracy declined monotonically as difficulty increased. At the same

Parameter	Pearson r	RMSE	MAE	Conv.
<i>SciEntsBank</i>				
grader ability (θ)	0.996	0.051	0.041	1.000
response difficulty (b)	0.791	0.595	0.461	1.000
<i>Beetle</i>				
grader ability (θ)	0.994	0.055	0.043	1.000
response difficulty (b)	0.786	0.595	0.464	1.000

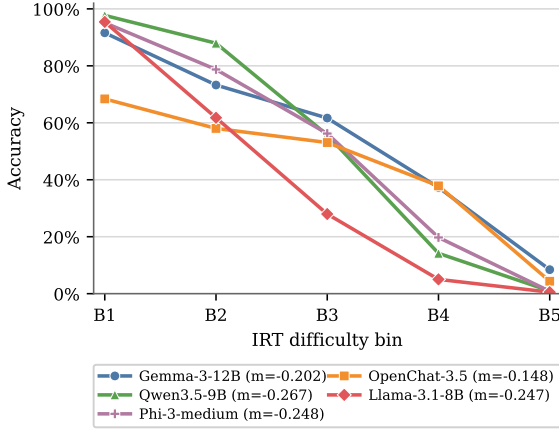
Table 2: Split-half stability results over 10 random replications on SciEntsBank and Beetle. Values are mean agreement statistics after mean–std alignment across halves. Conv. denotes convergence rate.

time, the rate of decline varied markedly across models, indicating substantial differences in robustness to difficult responses.

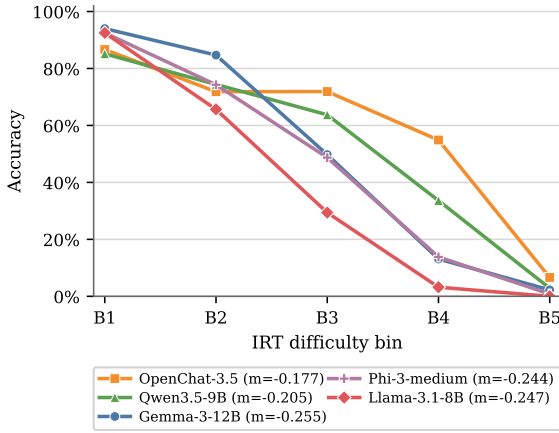
On SciEntsBank, all examined models showed sharp performance degradation toward the most difficult bins, but the extent of this decline differed. Gemma-3-12B remained the most robust overall, retaining 16.6% accuracy on $B5$, whereas Qwen3.5-9B, Phi-3-medium, OpenChat-3.5, and Llama-3.1-8B fell to 4.1%, 5.8%, 15.9%, and 2.1%, respectively. On Beetle, the same overall pattern held, although the relative robustness profile differed. OpenChat-3.5 degraded more gradually than the other models and retained the strongest performance in the upper-difficulty bins, reaching 67.2% on $B4$ and 12.9% on $B5$. By contrast, Gemma-3-12B and Phi-3-medium dropped much more sharply after $B3$, while Llama-3.1-8B approached near-zero performance already by $B4$.

Additional results for the full set of models are provided in Fig. 3 in Appendix A.2. Notably, the difficulty–performance slopes are broadly consistent across the two datasets. The cross-dataset association is strong, with a Pearson correlation of 0.838 and a Spearman correlation of 0.706. This suggests that sensitivity to increasing response difficulty is not merely dataset-specific, but instead reflects a comparatively stable characteristic of LLM grading behavior.

To further characterize this degradation, we examined confusion patterns across difficulty bins. As shown in Fig. 2 and Fig. 4 in Appendix A.2, errors in high-difficulty responses are not randomly distributed across labels, but instead become increasingly structured. Across both datasets, correct, contradictory, irrelevant, and non_domain responses are increasingly mapped to partially_correct_incomplete (PCI) as difficulty rises, indicating a progressive loss of fine-



(a) SciEntsBank



(b) Beetle

Figure 1: Model accuracy across ordered response-difficulty bins on (a) SciEntsBank and (b) Beetle. Here, m denotes the slope obtained by linearly regressing model accuracy on the order of the IRT-based difficulty bins.

grained multi-class discrimination. At the same time, difficult PCI responses are often misclassified as correct, suggesting that higher difficulty induces both collapse toward an intermediate category and overly optimistic judgments of ambiguous answers.

4.3 Semantic Misalignment and Response Atypicality as Correlates of Difficulty

As shown in Table 3, the correlation analysis reveals a clear and largely consistent pattern across SciEntsBank and Beetle. Overall, the strongest associations with response difficulty come from features that capture semantic alignment with the reference answer and the semantic typicality of a student response relative to other responses. In both datasets, higher difficulty is associated

with lower NLI entailment to the reference answer (SciEntsBank: Pearson $r = -0.300$, Spearman $\rho = -0.303$; Beetle: Pearson $r = -0.261$, Spearman $\rho = -0.227$) and a smaller entailment-contradiction margin (SciEntsBank: $r = -0.291$, $\rho = -0.306$; Beetle: $r = -0.235$, $\rho = -0.232$). Likewise, semantic similarity to the reference answer and lexical overlap with the reference answer, including both unigram and bigram overlap, are negatively correlated with difficulty in both datasets. These results indicate that responses that are semantically closer to the reference answer tend to be easier for LLM graders to evaluate correctly.

By contrast, features reflecting semantic disagreement or atypicality show positive associations with difficulty. In both datasets, the NLI contradiction score is positively correlated with response difficulty (SciEntsBank: $r = 0.211$, $\rho = 0.309$; Beetle: $r = 0.136$, $\rho = 0.216$). A similar pattern holds for embedding-space neighborhood distance. Both average and minimum k -nearest-neighbor distance are positively associated with difficulty across datasets. This suggests that responses that are more isolated in semantic space, and therefore less typical relative to the broader response distribution, are more difficult for LLM graders to grade reliably.

At the same time, not all features are equally informative. Some shallow surface-level properties, such as token count and type-token ratio, show weak or inconsistent associations across datasets. The number of missing reference segments is positively related to difficulty in SciEntsBank, but much weaker in Beetle.

5 Discussion

Our findings support a difficulty-aware perspective on evaluating LLM-based ASAG. Although aggregate measures such as accuracy, macro-F1, and agreement coefficients are useful summaries of overall performance, they do not reveal whether a model remains robust across responses of differing grading difficulty, a limitation also noted in prior IRT-based work on NLP evaluation (Lalor et al., 2016, 2018). The simulation-based recovery and split-half analyses further support the stability of the proposed measurement framework, suggesting that the estimated latent structure is not arbitrary and remains reasonably consistent across repeated analyses. Split-half results also show that grader ability estimates are more stable than response dif-

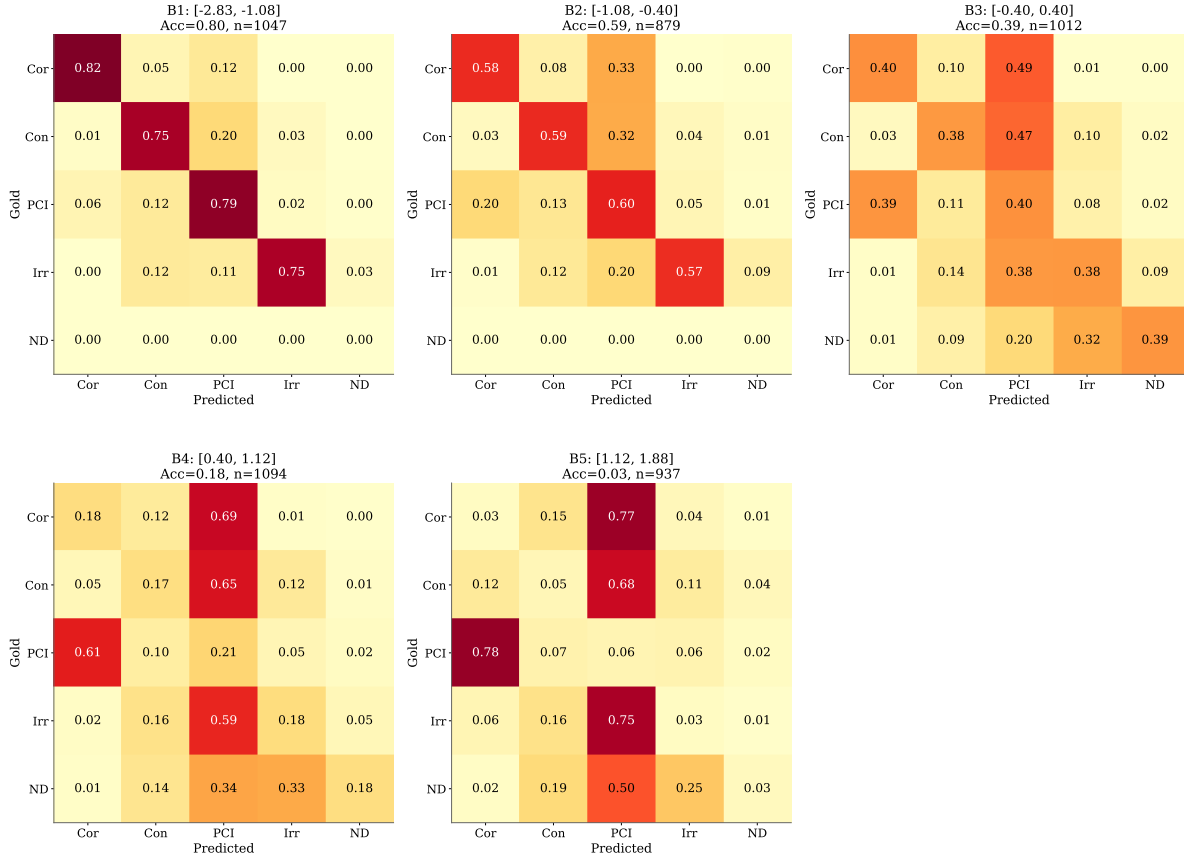


Figure 2: Confusion matrices on SciEntsBank across five bins of IRT-derived response difficulty b . n denotes the number of responses in each bin, and Acc denotes overall grading accuracy.

difficulty estimates, likely because the former are estimated from many responses, whereas the latter are supported by far fewer grader judgments.

This difficulty-aware analysis therefore provides a more informative account of where LLM graders succeed and where they fail. More specifically, the results suggest that LLM graders should be compared not only by their overall performance, but also by how robustly they handle increasingly difficult responses. Models with similar aggregate scores may nevertheless differ substantially in the rate and form of performance degradation across difficulty levels. From this perspective, a grader that performs well on easy cases but deteriorates sharply on difficult responses is qualitatively different from one whose performance declines more gradually, even when their aggregate scores are similar. From a robustness standpoint, the latter may be better suited for practical deployment.

The error analysis suggests that high-difficulty responses do not simply produce more mistakes at random. Instead, errors become increasingly concentrated in the `partially_correct_incomplete` category.

This pattern indicates that, under uncertainty, LLM graders often default to an intermediate label rather than making a sharper distinction. One plausible interpretation is that `partially_correct_incomplete` acts as a default intermediate category when evidence is weak, mixed, or only partially supportive. In such cases, the model preserves topical plausibility without committing to full correctness.

The feature analysis helps explain why such responses are difficult. Across datasets, higher difficulty is most consistently associated with weaker semantic alignment to the reference answer, stronger contradiction-related signals, and greater isolation in embedding space. These findings suggest that difficult responses are typically not arbitrary outliers, but responses whose meaning is incomplete, weakly expressed, internally mixed, or only partially compatible with the reference answer. At the same time, the correlations are moderate rather than large, which implies that difficulty is not reducible to any single response property. Instead, grading difficulty appears to emerge from the interaction of multiple semantic factors, including

Feature	SciEntsBank		Beetle	
	Pearson r	Spearman ρ	Pearson r	Spearman ρ
NLI entailment to reference	-0.300***	-0.303***	-0.261***	-0.227***
Entailment-contradiction margin	-0.291***	-0.306***	-0.235***	-0.232***
Average kNN distance	+0.206***	+0.250***	+0.210***	+0.200***
NLI contradiction to reference	+0.211***	+0.309***	+0.136***	+0.216***
Bigram overlap with reference	-0.247***	-0.222***	-0.147***	-0.110***
Semantic similarity to reference	-0.204***	-0.220***	-0.190***	-0.185***
Minimum kNN distance	+0.183***	+0.218***	+0.187***	+0.190***
Unigram overlap with reference	-0.203***	-0.192***	-0.160***	-0.127***
NLI neutrality to reference	+0.134***	+0.162***	+0.133***	+0.137***
Number of missing reference segments	+0.133***	+0.169***	+0.040	+0.027
Type-token ratio	+0.010	+0.050*	-0.055***	-0.050***
Token count	-0.008	-0.042	+0.050***	+0.064***

Table 3: Correlation between response difficulty and linguistic/semantic features on SciEntsBank and Beetle. Cell values are Pearson r and Spearman ρ ; superscripts indicate Benjamini-Hochberg adjusted significance (* $q < .05$, ** $q < .01$, *** $q < .001$). Features are ordered by mean absolute Pearson correlation across the two datasets.

alignment, contradiction, completeness, and local response-space structure.

Taken together, these results have practical implications for LLM-assisted grading. Most importantly, they suggest that not all responses should be treated as equally suitable for fully automatic scoring. If difficulty can be estimated directly or approximated from response features, it could support selective grading pipelines in which straightforward cases are handled automatically while difficult cases are routed to human review. Compared with uncertainty estimates derived solely from the model itself (Funayama et al., 2022; Bexte et al., 2024; Shorinwa et al., 2025), response-based difficulty also offers a more interpretable account of why particular answers are difficult to grade. This may be especially valuable in educational settings, where reliability, interpretability, and fairness are as important as efficiency (Madnani and Cahill, 2018).

The findings also have implications for rubric design, which has recently emerged as an important direction in ASAG research (Frohn et al., 2025; Cong et al., 2026b; Gombert et al., 2026). The systematic tendency to over-predict partially_correct_incomplete on difficult responses suggests that current prompting setups may not provide sufficiently clear decision boundaries between neighboring categories. More explicit rubric-based prompting or decomposed decision procedures may help address this problem. For example, grading could be structured into sequential sub-decisions such as topical relevance, presence of correct propositions, contradiction to the reference, and completeness of required content. This kind

of scaffolded evaluation may reduce the model’s tendency to collapse toward an intermediate class when evidence is ambiguous.

6 Conclusion

Overall, the present study shows that IRT offers a useful framework for analyzing LLM graders beyond aggregate performance. Rather than asking only which model achieves the highest average score, this perspective makes it possible to ask which responses are difficult, why they are difficult, and which models remain more robust as difficulty increases. For ASAG, this shift from average performance to measurement-oriented analysis provides a more interpretable basis for both evaluation and system design.

7 Limitations

This study has several limitations. First, the number of LLM graders is limited. Although we evaluate 17 models across multiple families and parameter scales, the estimated latent ability scale and robustness patterns should be interpreted with respect to this particular model set. Second, the analysis is restricted to two benchmark ASAG datasets and a specific five-way label schema. Although the main patterns are consistent across SciEntsBank and Beetle, it remains unclear how well they generalize to other domains, rubrics, or grading schemes. Third, our IRT formulation binarizes grading correctness and therefore does not model the five-way label structure directly. Future work could use polytomous IRT models to capture more fine-grained differences among label categories and error types. Finally, the feature analysis is correlational. The

observed associations help characterize difficult responses, but they do not establish causal determinants of grading difficulty. Investigating these causal mechanisms remains an important direction for future work.

Acknowledgments

This research was conducted within the project “Assessment for Learning with AI (ALwAI)” funded by the Leibniz Association under the Leibniz Competition (project no. T163/2024).

References

- 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.
- Marie Bexte, Andrea Horbach, Lena Schützler, Oliver Christ, and Torsten Zesch. 2024. Scoring with confidence?—exploring high-confidence scoring for saving manual grading effort. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 119–124.
- Sridevi Bonthu, S Rama Sree, and MHM Krishna Prasad. 2021. Automated short answer grading using deep learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pages 61–78. Springer.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International journal of artificial intelligence in education*, 25(1):60–117.
- Leon Camus and Anna Filighera. 2020. Investigating transformers for automatic short answer grading. In *International conference on artificial intelligence in education*, pages 43–48. Springer.
- Jianhao Chen, Chenxu Wang, Gengrui Zhang, Peng Ye, Lei Bai, Wei Hu, Yuzhong Qu, and Shuyue Hu. 2025. Learning compact representations of llm abilities via item response theory. *arXiv preprint arXiv:2510.00844*.
- Junhyuk Choi, Sohhyung Park, Chanhee Cho, Hyeonchu Park, and Bugeun Kim. 2026. Diagnosing the reliability of llm-as-a-judge via item response theory. *arXiv preprint arXiv:2602.00521*.
- Longwei Cong, Sonja Hahn, Sebastian Gombert, Leon Camus, Hendrik Drachsler, and Ulf Kroehne. 2026a. [Confidence estimation in automatic short answer grading with llms](#). *Preprint*, arXiv:2605.00200.
- Longwei Cong, Leon Hammerla, Sonja Hahn, Sebastian Gombert, Hendrik Drachsler, and Ulf Kroehne. 2026b. [Automatic short answer grading with LLMs: From memorization to reasoning](#). In *Proceedings of the 16th International Learning Analytics and Knowledge Conference*, New York, NY, USA. Association for Computing Machinery.
- 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.
- Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. 2025. Automatic short answer grading in the llm era: Does gpt-4 with prompt engineering beat traditional models? In *Proceedings of the 15th international learning analytics and knowledge conference*, pages 93–103.
- Scott Frohn, Tyler Burleigh, and Jing Chen. 2025. Automated scoring of short answer questions with large language models: Impacts of model, item, and rubric design. In *International Conference on Artificial Intelligence in Education*, pages 44–51. Springer.
- Hiroaki Funayama, Tasuku Sato, Yuichiro 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.
- Sebastian Gombert, Daniele Di Mitri, Onur Karademir, Marcus Kubsch, Hannah Kolbe, Simon Tautz, Adrian Grimm, Isabell Böhm, Knut Neumann, and Hendrik Drachsler. 2023. Coding energy knowledge in constructed responses with explainable nlp models. *Journal of Computer Assisted Learning*, 39(3):767–786.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berrit Katharina Czinczel, David Bednorz, Marcus Kubsch, Daniele Di Mitri, Knut Neumann, and Hendrik Drachsler. 2026. Are rubrics all you need? towards rubric-based automatic short answer scoring via guided rubric–answer alignment. In *Proceedings of the 16th International Learning Analytics and Knowledge Conference*.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. [A survey on llm-as-a-judge](#). *Preprint*, arXiv:2411.15594.

- Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. 2022. Survey on automated short answer grading with deep learning: from word embeddings to transformers. *arXiv preprint arXiv:2204.03503*.
- Ronald K Hambleton and Hariharan Swaminathan. 2013. *Item response theory: Principles and applications*. Springer Science & Business Media.
- Michael R Harwell. 1997. Analyzing the results of monte carlo studies in item response theory. *Educational and Psychological Measurement*, 57(2):266–279.
- John P. Lalor, Hao Wu, Tsendsuren Munkhdalai, and Hong Yu. 2018. [Understanding deep learning performance through an examination of test set difficulty: A psychometric case study](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4711–4716, Brussels, Belgium. Association for Computational Linguistics.
- John P. Lalor, Hao Wu, and Hong Yu. 2016. [Building an evaluation scale using item response theory](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657, Austin, Texas. Association for Computational Linguistics.
- Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhat-tacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. [From generation to judgment: Opportunities and challenges of LLM-as-a-judge](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.
- Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.
- Nitin Madnani and Aoife Cahill. 2018. [Automated scoring: Beyond natural language processing](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1099–1109, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. [Learning to grade short answer questions using semantic similarity measures and dependency graph alignments](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA. Association for Computational Linguistics.
- Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 567–575.
- 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.
- Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. [Evaluation examples are not equally informative: How should that change NLP leaderboards?](#) 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 4486–4503, Online. Association for Computational Linguistics.
- Alexander Scarlatos, Nigel Fernandez, Christopher Ormerod, Susan Lottridge, and Andrew Lan. 2025. Smart: Simulated students aligned with item response theory for question difficulty prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25082–25105.
- Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. 2025. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*, 58(3):1–38.
- Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. [Fast and easy short answer grading with high accuracy](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075, San Diego, California. Association for Computational Linguistics.
- Wim J. van der Linden. 2016. *Handbook of item response theory*, volume 1. CRC press Boca Raton, FL.
- Howard Wainer, Eric T Bradlow, and Xiaohui Wang. 2007. *Testlet response theory and its applications*. Cambridge University Press.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.
- Hongli Zhou, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, and 1 others. 2025. Lost in benchmarks? rethinking large language model benchmarking with item response theory. *arXiv preprint arXiv:2505.15055*.

A Appendix

A.1 Feature Definitions

For each student response, we compute a set of lexical, semantic, and distributional features with respect to the reference answer and the question. Let a_i denote the student answer for item i , r_i the corresponding reference answer, and q_i the question. All lexical features are computed from lower-case regex-based tokens extracted with the pattern $[A-Za-z0-9']^+$. Thus, throughout this appendix, “token” refers to these surface tokens rather than model-specific subword tokens.

Lexical overlap features. Let $U(a_i)$ and $U(r_i)$ denote the sets of unique unigrams in the student answer and reference answer, respectively. The unigram overlap with the reference is defined as

$$\text{UnigramOverlap}(i) = \frac{|U(a_i) \cap U(r_i)|}{|U(r_i)|}.$$

Similarly, letting $B(a_i)$ and $B(r_i)$ denote the sets of unique bigrams in the student answer and reference answer, the bigram overlap is

$$\text{BigramOverlap}(i) = \frac{|B(a_i) \cap B(r_i)|}{|B(r_i)|}.$$

Both measures therefore quantify reference-normalized lexical coverage rather than Jaccard similarity.

The token count is simply the number of tokens in the student answer,

$$\text{TokenCount}(i) = |a_i|,$$

and the type-token ratio is defined as

$$\text{TTR}(i) = \frac{|\text{unique tokens in } a_i|}{|a_i|}.$$

Semantic similarity features. We encode student answers, reference answers, and questions using Sentence-BERT embeddings. All embeddings are ℓ_2 -normalized, so cosine similarity reduces to a dot product. The semantic similarity to the reference is computed as

$$\text{SimRef}(i) = \cos(\mathbf{e}(a_i), \mathbf{e}(r_i)).$$

NLI-based features. We apply a pretrained natural language inference model to each reference-answer pair, using the reference answer as premise

and the student answer as hypothesis. This yields three probabilities:

$$P_{\text{ent}}(i), \quad P_{\text{con}}(i), \quad P_{\text{neu}}(i),$$

corresponding to entailment, contradiction, and neutrality. From these, we derive the following features:

$$\begin{aligned} \text{NLIEntail}(i) &= P_{\text{ent}}(i), \\ \text{NLIContradict}(i) &= P_{\text{con}}(i), \\ \text{NLINeutral}(i) &= P_{\text{neu}}(i), \\ \text{NLIMargin}(i) &= P_{\text{ent}}(i) - P_{\text{con}}(i). \end{aligned}$$

Distributional neighborhood features. To quantify how typical or atypical a student response is relative to other responses in the same dataset, we compute pairwise cosine similarities among all student-answer embeddings. For each answer, we identify its k nearest neighbors in embedding space (excluding the answer itself), with $k = 5$ in our implementation. The average k NN distance is defined as

$$\text{AvgKNNDist}(i) = 1 - \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \cos(\mathbf{e}_i, \mathbf{e}_j),$$

where $\mathbf{e}_i = \mathbf{e}(a_i)$ denotes the embedding of answer a_i , and $\mathcal{N}_k(i)$ denotes the set of indices of the k most similar other student answers. The minimum k NN distance is defined as

$$\text{MinKNNDistance}(i) = 1 - \max_{j \neq i} \cos(\mathbf{e}(a_i), \mathbf{e}(a_j)).$$

Higher values indicate that a response is more isolated in the embedding space.

Reference-segment coverage. To approximate coverage of distinct parts of the reference answer, we split each reference answer into segments using sentence-like punctuation (. ; : ! ?), tokenize each segment, and retain only segments with at least three tokens. For each retained segment s , we compute its token overlap with the set of answer tokens:

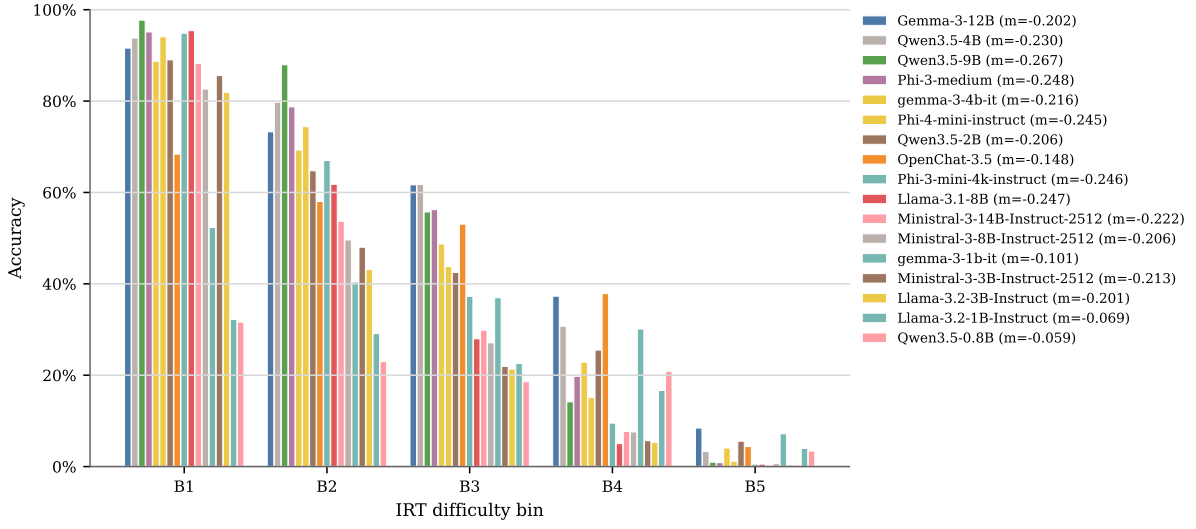
$$\text{Coverage}(s, a_i) = \frac{|U(s) \cap U(a_i)|}{|U(s)|}.$$

A reference segment is considered covered if this value is at least 0.5. The number of missing reference segments is then

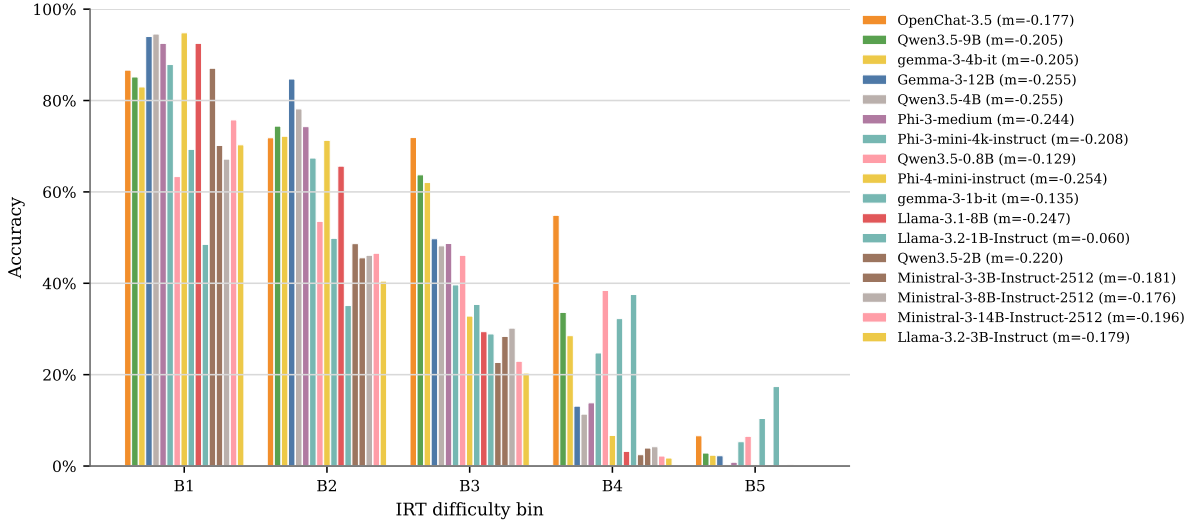
$$\begin{aligned} \text{MissingSegments}(i) &= N_{\text{segments}}(r_i) \\ &\quad - N_{\text{covered}}(r_i). \end{aligned}$$

Higher values indicate that the student answer fails to cover more reference segments.

A.2 Additional Plots and Tables



(a) SciEntsBank



(b) Beetle

Figure 3: Model accuracy across ordered IRT-based difficulty bins for all evaluated models on (a) SciEntsBank and (b) Beetle. Here, m denotes the slope obtained by linearly regressing model accuracy on the order of the IRT-based difficulty bins.

LLMs	SciEntsBank			Beetle		
	θ	Acc	Macro-F1	θ	Acc	Macro-F1
google/gemma-3-12b-it	0.805	0.546	0.474	0.487	0.494	0.463
Qwen/Qwen3.5-4B	0.758	0.538	0.434	0.404	0.470	0.314
Qwen/Qwen3.5-9B	0.575	0.508	0.402	0.653	0.518	0.395
microsoft/Phi-3-medium-4k-instruct	0.545	0.499	0.428	0.330	0.464	0.375
google/gemma-3-4b-it	0.367	0.467	0.377	0.515	0.496	0.434
microsoft/Phi-4-mini-instruct	0.309	0.455	0.322	0.084	0.416	0.253
Qwen/Qwen3.5-2B	0.285	0.455	0.350	-0.463	0.322	0.172
openchat/openchat-3.5-0106	0.264	0.446	0.356	0.963	0.577	0.494
microsoft/Phi-3-mini-4k-instruct	0.117	0.416	0.335	0.197	0.450	0.414
meta-llama/Llama-3.1-8B-Instruct	-0.100	0.379	0.224	-0.125	0.386	0.338
mistralai/Mistral-3-14B-Instruct-2512	-0.211	0.359	0.269	-0.645	0.296	0.246
google/gemma-3-1b-it	-0.301	0.337	0.250	-0.075	0.391	0.273
mistralai/Mistral-3-8B-Instruct-2512	-0.349	0.335	0.242	-0.649	0.297	0.282
mistralai/Mistral-3-3B-Instruct-2512	-0.398	0.323	0.181	-0.645	0.298	0.264
meta-llama/Llama-3.2-3B-Instruct	-0.491	0.304	0.148	-0.798	0.266	0.161
meta-llama/Llama-3.2-1B-Instruct	-1.053	0.209	0.144	-0.342	0.331	0.188
Qwen/Qwen3.5-0.8B	-1.122	0.197	0.132	0.109	0.412	0.193

Table 4: Latent grading ability estimates (θ), accuracy, and macro-F1 for LLM judges on SciEntsBank and Beetle under the IRT model.

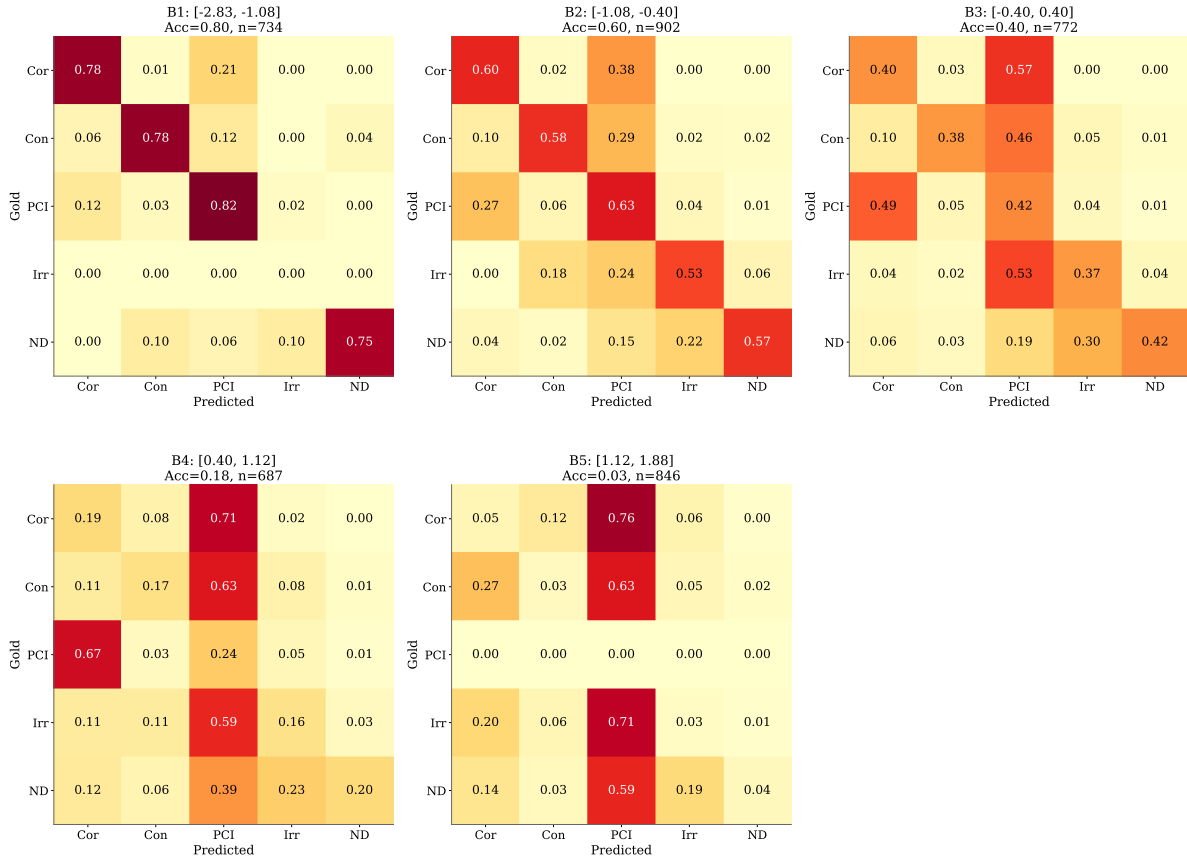


Figure 4: Confusion matrices on Beetle across five bins of IRT-derived response difficulty b . n denotes the number of responses in each bin, and Acc denotes overall grading accuracy.

Using k -Shot Prompting with Large k for the Automated Scoring of a German Written Elicited Imitation Test

Malte Sternik¹, Ronja Laarmann-Quante¹ and Anastasia Drackert^{2,3}

Ruhr University Bochum, Germany
firstname.lastname@rub.de

¹Faculty of Philology, Department of Linguistics

²Faculty of Philology, Institute for German Language and Literature

³g.a.s.t. (Society for Academic Study Preparation and Test Development)

Abstract

This paper explores the application of a Large Language Model (LLM) using k -shot prompting with large k for automatically scoring a German Written Elicited Imitation Test (WEIT), a test for assessing literacy-dependent procedural knowledge in German as a foreign language. In this test, test-takers are briefly presented with written sentences which they then have to reproduce in writing as accurately as possible. The responses are scored on an ordinal scale which differentiates between different types of errors (e.g. lexical vs. grammatical). We find that with increasing k (in a range from 1 to 700) accuracy increases significantly but it also depends on the drawn sample and varies across different runs of the same prompt. Overall, the k -shot setting which relies on in-context learning without being provided with the scoring rubric outperforms a baseline where only the scoring rubric is provided to the model. However, the LLM does not outperform previous results based on rule-based or BERT-based models.

1 Introduction

The Written Elicited Imitation Test (WEIT) is a test used for measuring procedural linguistic knowledge in writing (Timukova et al., 2026). In this computer-based test, participants view sentence after sentence for a short time duration and after a brief pause, they are asked to reproduce the sentence by typing it in a text field. The responses are rated on an ordinal scale ranging from 0 to 4 based on a scoring rubric. Responses receive higher scores when they are closer to the original while lower scores are given to grammatically, syntactically or semantically altered sentences (see Section 3 for a more detailed description of the scoring rubric).

Chiffigarov et al. (2025) were the first to explore the automatic scoring of a German WEIT. They compared two approaches: (i) a rule-based scoring model which directly implements categories

from the scoring rubric and (ii) a deep-learning-based approach where a pre-trained transformer model from the BERT family (DistilBERT; Sanh et al., 2019), henceforth referred to as BERT-based model, was finetuned on pairs of annotated stimulus and response sentences in order to learn (implicitly) which scores to apply. The results showed that both approaches had different strengths and weaknesses. While the rule-based model performed better on unseen stimulus sentences and scores at the edges of the rating scale, the BERT-based model was better at mid-range scores.

Overall, the results appear not yet robust enough to be applied in practice on the level of the individual response. At the same time, language learners and instructors would greatly benefit from an automatic scoring of WEITs, which would reduce the manual evaluation effort and thus make this instrument more widely usable. Developing a more robust scoring model is thus still a desideratum.

The present paper is a follow-up to the study of Chiffigarov et al. (2025), exploring the use of a Large Language Model (LLM) for automatically scoring a German WEIT. It focuses on one specific prompting strategy and systematically investigates different sources of variance connected to this strategy. The strategy we use is k -shot prompting, an in-context learning approach where the LLM is presented with k scored samples and in our case no explicit scoring rubric. The idea is that based on the provided examples (shots), the LLM derives the scoring principles on its own. Besides the choice of k , we study the impact of different samples drawn. Another source of variance applying to all LLM-based studies is the non-determinism of an LLM, i.e. different runs of the same prompt can produce different results even if the temperature parameter is set to 0 (Atil et al., 2025). This source of variance has rarely been studied systematically (Song et al., 2025, but see e.g. Ouyang et al., 2025).

The **aim of the paper** is twofold: (1) compar-

ing the performance of an LLM-based model with the rule-based and BERT-based models used by Chiffigarov et al. (2025), thereby contributing to the development of automatic scoring models for WEITs and (2) an in-depth investigation of the k -shot prompting strategy used for the LLM, contributing to the understanding of variability in LLM performance for this particular task. We pursue the following **research questions** related to (2):

RQ1: What is the optimal number of examples k for scoring a WEIT?

RQ2: What variance in performance arises through the use of different samples for the in-context learning examples?

RQ3: Given the non-determinism of LLMs, what variance is to be expected only by running the same prompt multiple times?

The LLM used in this paper is GPT-4o by OpenAI, which was chosen due to its popularity. Section 2 discusses related work and Section 3 presents the data and scoring rubric. In Experiment 1 (Sec. 4), we explore a large range of k ($10 \leq k \leq 700$) while in Experiment 2 (Sec. 5), we focus on small $k \leq 25$. In Section 6 we compare the results to those obtained by Chiffigarov et al. (2025). The code and data from this study are available on <https://gitlab.ruhr-uni-bochum.de/vamos-cl/german-weit-k-shot-prompting>.

2 Related Work

Chiffigarov et al. (2025) were the first to automatically score a written EIT. Previously, only the automated scoring of oral EITs (OEIT) had been explored. However, the OEITs were typically rated on a binary scale (Millard, 2011) or interval scale (Graham et al., 2008; Lonsdale and Christensen, 2011), not an ordinal scale with a rubric that distinguishes different error categories (see Sec. 3).

In other areas of the educational domain involving automated scoring, LLMs have already become popular. Different studies investigated the use of k -shot prompting, especially few-shot prompting, coming to mixed results. In the domain of automatic essay scoring, for example, Mansour et al. (2024) found that adding one example (1-shot prompting) to the scoring rubric improves the results but not across all models and all writing tasks. Yoshida (2024) investigated 1–3 shots, finding that the choice and order of examples impacts the results. Huang and Wilson (2025) found that few-shot prompting improves the results. They added

five examples to a prompt providing context and chain-of-thought instructions. In the area of short-answer scoring, Chamieh et al. (2024) explored k -shot prompting with k varying between 1 and 10. They found that increasing the number of shots was not beneficial for all models and datasets.

For other domains, there has already been evidence that in-context learning with a large k can be powerful. Bertsch et al. (2025) used long context models on different classification tasks, e.g. question classification or intent classification. They found accuracy gains of up to 50.8 points when scaling from 10 to 1,000 examples. Furthermore, with increasing k , it became less important which examples were chosen. Note that their prompts *only* contained the shots. In our experiments, we follow this approach of using large values of k and, while we provide a little bit of context in the prompt, we do not include the full scoring rubric either, which means that the model predominantly relies on the provided examples. In this respect, our approach differs from the k -shot prompting studies in the areas of short-answer scoring or essay scoring discussed above, where a rather small number of shots (≤ 10) was used to support rather than replace the scoring guidelines in the prompt. Furthermore, in the context of a WEIT, a single shot contains only one sentence pair, whereas for essay scoring, a single shot is equivalent to a whole text.

3 Data

We use the WEIT dataset published by Chiffigarov et al. (2025). Its collection, the creation of a scoring rubric and the manual scoring of the data was part of a larger research project (Timukova et al., 2026). The dataset consists of 3,900 stimulus-response pairs (195 participants responded to 20 different stimulus sentences each). Each response was scored based on an ordinal scoring rubric. In the following, we quote a scored example (Table 1) and the brief summary of the scoring rubric from Chiffigarov et al. (2025). The full rubric can be found in their supplementary material.

Score 4 The response matches the stimulus sentence exactly or 1–2 typos are present.¹

Score 3 Changes in grammar or lexical changes that preserve the original structure and result

¹Typos include: transposed letters (all present), one letter replaced by a QWERTZ-adjacent key, one letter added/omitted next to an adjacent key, or a missing space between words.

Score	Example	Explanation
0	Bein praktikum	less than half of the words repeated correctly
1	Bei einem Praktikum * * *	half of the words repeated correctly but most of the meaning lost
2	Bei ein Praktikum lernt man viel.	case wrongly marked, ungrammatical sentence
3	Beim Praktikum lernt man viel.	change in grammar (contraction of preposition + article) but still grammatical
4	Bei einem Praktikum lenrt man viel.	one typo

Table 1: Example of the scoring rubric for the stimulus sentence *Bei einem Praktikum lernt man viel*. ‘At an internship, one learns a lot’. Table taken from [Chiffigarov et al. \(2025\)](#).

in grammatically correct and meaningful sentences, e.g. confusing definite and indefinite articles (where interchangeable), or (near-)synonymic substitutions of words.

Score 2 Changes in grammar that result in ungrammatical sentences or grammatical sentences which are not meaningful, e.g. violated agreement between subject and verb, structural omissions or wrong plural formation.

Score 1 More than half of the words are repeated but a considerable part of the original meaning or structure is lost or changed.

Score 0 Less than half of the words are repeated.

We use the same split into training, development and test data as used by [Chiffigarov et al. \(2025\)](#).

4 Experiment 1: Large Samples

In the following experiments, the LLM GPT-4o (henceforth referred to as GPT model) is prompted to score the data described in Section 3 only based on examples from the training data, without explicitly providing the scoring rubric. In Experiment 1, three sources of variance are explored: (1) the number of examples k , (2) different samples drawn from the training set (consisting of 3,260 stimulus-response pairs in total) (3) different runs of the same prompt using the same examples.

4.1 Method

We vary the number of examples k between 10 and 700. Between $k = 10$ and $k = 50$, k is increased in steps of 10, above $k = 50$ in steps of 50, yielding 18 variations of k in total. For each k , two different samples (based on a fixed random seed) are drawn from the training set. For each k and each sample, four runs of the same prompt are carried out, yielding a total of 144 runs (18 different $k \times 2$ different samples $\times 4$ runs). To keep the variance as low as possible, we set the temperature parameter to 0.

Identity:

You score answers from a German Elicited Imitation test. You score between 0 and 4. The test is in german language.

Instructions:

- The training data contains answers and the original items, that are already scored between 0 and 4
- The training data has an ID at the beginning, the original second, the answer as third and the score as fourth
- Try to understand why and how the scores of the training data relate to the answer original pairs
- The training data is here: **EXAMPLES**
- You will get the same answer original pairs but with no scores, score each pair based on the training data.
- Output only the answer original pairs with the respective scores and ID of the answer original pair
- Format the output in a way that the ID, the answer, the original and the score are divided by ‘\t’

Figure 1: Prompt given to GPT-4o in the developer/system role.

Prompt The model is given a set of stimulus-response pairs to score from the user role. A simple instruction prompt from the developer/system role is added, asking the model to analyze the given examples (termed training data), see Figure 1. The examples are provided as a simple string with a new line for each stimulus-response pair and divided into four columns named *id*, *original* (=stimulus), *answer* (=response) and *score* divided by a tab. The string is included within the prompt (marked by **EXAMPLES** in Figure 1). A little bit of manual adjusting of the prompt was done in order to get the model to just output the scored items and not add an unwanted information (reasons, questions, comments etc.). We are aware that subtle prompt formatting, e.g. regarding casing or spacing can impact the result ([Sclar et al., 2024](#)). Investigating this systematically was beyond the scope of this study though.

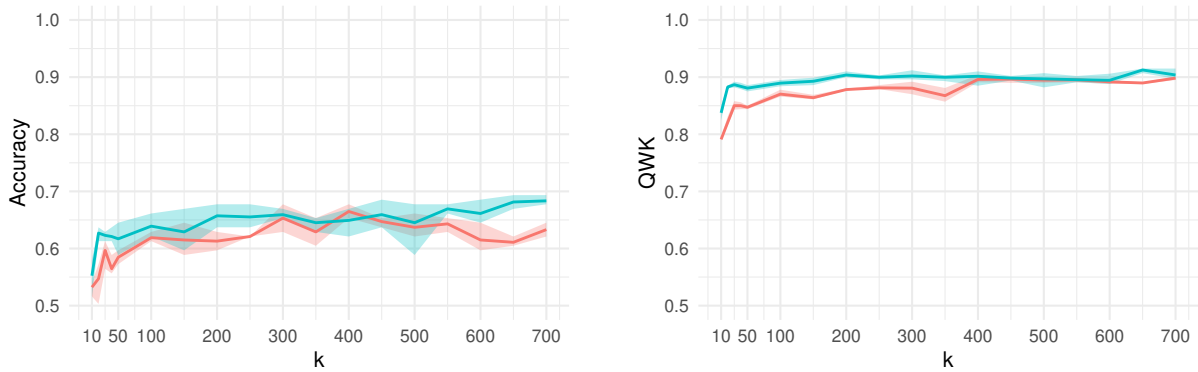


Figure 2: Results for **Experiment 1**: Accuracy (left) and QWK (right) for each number of shots k . The red and blue line represent the two different samples. The solid line represents the mean of the four runs, the shaded area indicates the minimum and maximum value per run.

Evaluation The results are evaluated on the development set consisting of 125 stimulus-response pairs that are balanced across scores. We use accuracy and quadratically weighted kappa (QWK) as evaluation metrics. While accuracy is a strict metric, only counting exact score matches, QWK quantifies the magnitude of deviations, punishing greater deviations from the gold score more severely than smaller deviations.

4.2 Result

Figure 2 illustrates how accuracy and QWK vary with the number of shots k , the chosen sample and the different runs. Overall, QWK is rather stable across k , especially with $k > 50$, and there is almost no influence of the different runs. When $k < 400$, there is a notable difference between the two samples. Accuracy shows much more variability involving all factors. There is a clear drop in performance with very low values of k , which will be investigated more closely in Experiment 2 in Section 5.

In order to assess the influence of the different factors, we conducted a binomial logistic regression analysis with accuracy as the dependent variable and k (numeric), *sample* (categorical) and *run* (categorical) and the interaction of k and *sample* as predictors.² Rather than modeling *sample* and *run* as random effects in a generalized linear mixed-effects model, which would conceptually correspond better with their randomness, we decided to include them as fixed effects since we wanted to study whether they have a systematic influence on scoring accuracy. Furthermore, their small number

of factor levels (two for *sample* and four for *runs*) does generally not make them suitable random effects (Gomes, 2022). In our analysis, the variable *run* was effects-coded so that the coefficients reflect deviations from the overall mean of the dependent variable (in contrast to the default treatment coding, where the comparison is made against one reference level of the predictor). We found two significant effects, namely the main effects of k ($\beta < .01$, $SE < .01$, $z = 4.46$, $p < .001$), and *sample* ($\beta < .13$, $SE < .05$, $z = 2.54$, $p = .01$). This means that a higher k yields a higher accuracy and also one of the samples yields a significantly higher accuracy than the other.

One aspect that certainly influences the performance of the models is whether they have seen all 18 stimulus sentences in the examples (see also Section 6 for the performance on known vs. unknown stimulus sentences). In fact, only with $k \geq 100$, all stimulus sentences are present in the examples. To investigate this further, we conducted two separate regression analyses on the data with $k < 100$ and $k \geq 100$. In these separate regressions, none of the factors are significant anymore (all $\beta < .18$, $SE < .14$, $-.64 < z < 1.8$, $p > .08$). This indicates that the significant effects of k and *sample* arise only when considering the whole range of k between 10 and 700.

5 Experiment 2: Small Samples

In Experiment 1, we found that overall, a higher number of shots k improves the accuracy and QWK and that there is a notable drop in performance when only few examples are given. To investigate this further, Experiment 2 uses even smaller values of k .

²The regression was carried out using the `glm` function of R (version 4.5.3)

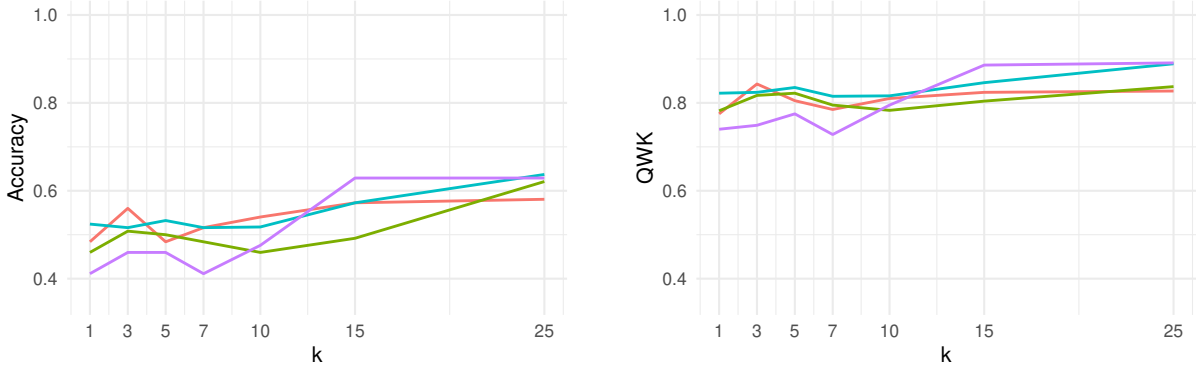


Figure 3: Results for **Experiment 2**: Accuracy (left) and QWK (right) for each number of examples k . Each line represents a different sample.

5.1 Method

In this experiment, $k \in \{1, 3, 5, 7, 10, 15, 25\}$ is used. We assume that the smaller the number of examples, the more it matters which ones are chosen. We therefore compare 4 different samples drawn from the training data with a fixed random seed per value of k . Since the different runs were not a significant factor in Experiment 1, we run each of the 28 configurations in this experiment (7 variations of $k \times 4$ different samples) only once. The prompt and the evaluation setting are the same as in Experiment 1.

5.2 Result

Figure 3 illustrates the accuracy and QWK for the different values of k and the four samples. As expected, for small k values, the chosen sample can make a big difference of over ten percentage points in accuracy. It is notable, though, that even with very few examples, the GPT model performs way above chance level (which would be 20% percent in the balanced development set on which the evaluation is carried out).

We again conducted a binomial logistic regression analysis with accuracy as the dependent variable and k (numeric) and *sample* (categorical) and their interaction as predictors. The variable *sample* was effects-coded. Only k turns out as a significant predictor ($\beta = .02, SE = .004, z = 5.53, p < .001$), with higher k yielding higher accuracy. Higher values of k are strongly correlated with the number of stimulus sentences seen (Pearson’s $r = 0.95$), which certainly influences model performance.

Model	Known It.		Unknown It.		Combined	
	acc	qwk	acc	qwk	acc	qwk
Rule-based	.71	.89	.60	.77	.64	.82
BERT	.77	.93	.58	.81	.64	.86
GPT-rubric	.51	.72	.46	.74	.47	.73
GPT-kshot	.66	.89	.55	.80	.59	.84

Table 2: Accuracy and QWK for the rule-based and BERT-based models from Chiffigarov et al. (2025) compared to the GPT-rubric and GPT-k-shot model on the test sets with known items and unknown items, respectively, and the combined test set. Numbers in bold indicate the best model per set and metric.

6 Comparison with Rule-Based and BERT-Based Models

To assess the overall performance of the GPT k -shot model, we compare it to the rule-based and BERT-based models from Chiffigarov et al. (2025).³ In this evaluation, we use the k -shot model that performed best on the development set in Experiment 1 ($k = 700$). Furthermore, we add another GPT-based baseline, where the prompt for GPT-4o contains the scoring rubric used by the human raters (henceforth referred to as GPT-rubric). In this setting, no scored examples (shots) are given but the rubric contains some concrete examples illustrating each score. The exact prompt with the rubric can be found in the online repository. Systematic tuning of this rubric-based prompt was beyond the scope and goal of this paper, so this setting is only meant as a basic point of reference for comparing the k -shot setting to.

For the evaluation, we use the two test sets from Chiffigarov et al. (2025). The first test set contains a

³In the following, the BERT-based model refers to the weighted model used by Chiffigarov et al. (2025).

Score			Precision				Recall				F1-Score			
	# answ.	%	Rule	BERT	GPT-r	GPT-k	Rule	BERT	GPT-r	GPT-k	Rule	BERT	GPT-r	GPT-k
0	92	24%	.85	.84	.63	.86	.91	.82	.29	.55	.88	.83	.40	.68
1	99	26%	.51	.66	.41	.54	.86	.61	.44	.64	.64	.63	.43	.59
2	77	20%	.29	.47	.28	.40	.17	.74	.26	.30	.21	.57	.27	.34
3	79	21%	.80	.58	.50	.57	.35	.42	.71	.68	.49	.49	.58	.62
4	34	9%	.94	.82	.72	.62	.94	.53	.97	.97	.94	.64	.82	.76
macro avg			.68	.67	.51	.60	.65	.62	.54	.63	.63	.63	.50	.60
micro avg			.64	.66	.48	.60	.64	.64	.47	.59	.61	.64	.46	.58

Table 3: Distribution of (gold) scores in the combined test set and precision, recall and F1-score per (gold) score as well as macro average and micro (=weighted) average for each model (GPT-r = GPT-rubric, GPT-k = GPT-k-shot). The numbers in bold indicate the highest value in precision, recall, and F1-score, respectively, per score.

random held-out sample of answers comprising all 18 different stimulus sentences that are also part of the training and development set (henceforth *known test set*). This test set reflects the performance on new answers to known stimulus sentences. It is balanced across all scores 0-4. The second test set (*unknown test set*) contains all answers to two completely new stimulus sentences, reflecting the performance of the models when a new test with new stimuli is used.

In contrast to Chiffigarov et al. (2025), we remove duplicates from the test set. Duplicate answers mainly concern answers that are practically identical to the stimulus sentence, therefore mostly answers with the highest score 4 are removed. Furthermore, due to a token limit for the GPT model, not all answers from the unknown test set could be processed but only a random sample. Consequently, our test set comprises the intersection of answers processed by all models and excluding the duplicates. This yields 116 answers (out of 125) in the known test set and 265 (out of 390) in the unknown test set.

Table 2 shows accuracy and QWK for each model for the known test set, the unknown test set and the combination of both test sets.⁴ We can see that GPT-k-shot outperforms GPT-rubric on all test sets and the difference in correctly scored items is significant on both sets (both $\chi^2(1) > 4.71$, $p < .03$, two-sided). However, GPT-k-shot does not perform better than the models presented in Chiffigarov et al. (2025), neither on the known nor on the unknown set. In fact, the differences in correctly

scored items between GPT-k-shot on the one hand and the rule-based and BERT-based model on the other hand are not significant (all $\chi^2(1) < 2.57$, $p > .10$, two-sided).

All models tend to perform worse on the unknown than on the known test set. The performance drop is greater for the BERT-based model than for the GPT-k-shot model although both models rely on training samples. However, the number of correctly scored items only differs significantly for the BERT-based model ($\chi^2(1) = 11.31$, $p < .001$, two-sided) not for the other models (all $\chi^2(1) < 3.53$, $p > .06$, two-sided). The GPT-rubric model has the smallest performance drop. This result is intuitive because the scoring rubric describes more general scoring rules applying to all stimulus sentences. Still, GPT-rubric shows the worst performance on the unknown test set compared to the other models.

Table 3 breaks down the results, showing precision, recall and F1-score per score for each model on the combined test set. We see that due to duplicate removal, score 4 only applies to 9% of the answers, according to the gold standard, whereas the other scores are more evenly distributed. As already noted by Chiffigarov et al. (2025), the rule-based model outperforms the other models on the edges of the rating scale, i.e. scores 0, 1 and 4, whereas the BERT-based model clearly performs better than the others on score 2. The rule-based model does not perform so well on scores 2 and 3, partly because not all categories from the scoring rubric pertaining to these scores have been implemented as they are not straightforward to be captured in rules. On score 3, GPT-k-shot has the best overall performance based on F1-score, although the rule-based model has a higher precision and GPT-rubric a slightly higher recall. As mentioned in Section 3, score 2 mainly captures grammatical

⁴Note that in Table 2, the performance of the rule-based model is slightly worse and that of the BERT-based model slightly better than reported in Chiffigarov et al. (2025). This is mainly due to the removal of duplicates, which drastically reduces the number of answers with score 4, a category where the rule-based model outperforms the BERT-based model.

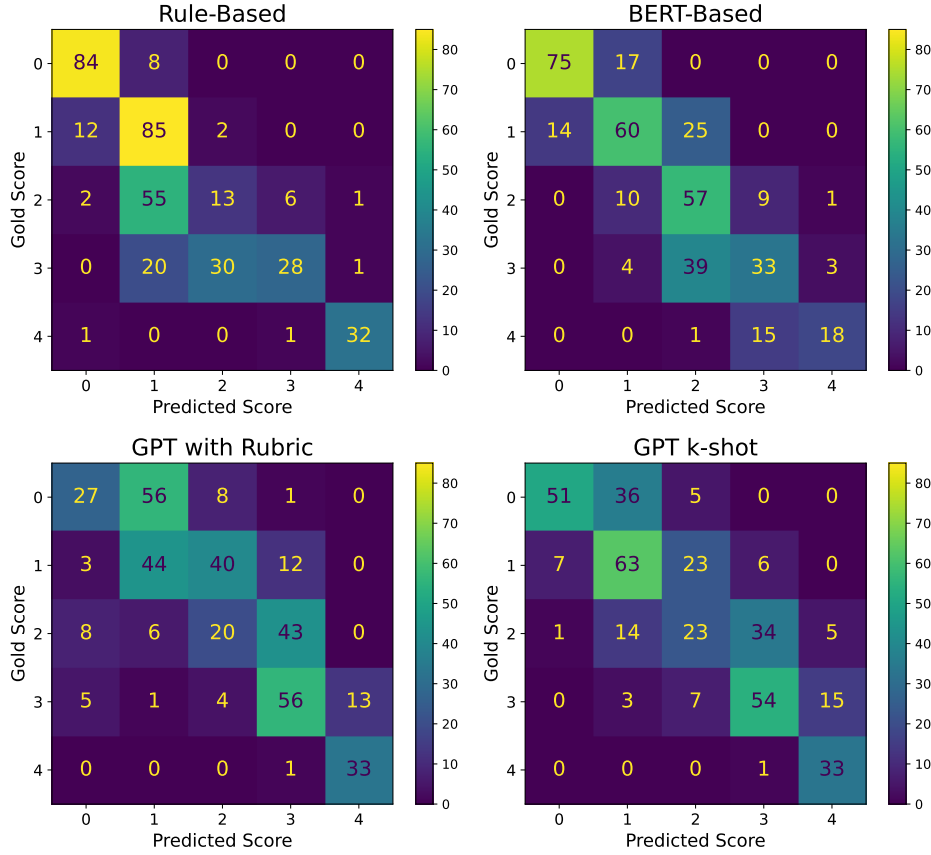


Figure 4: Confusion matrix of predicted vs. gold score for each model on the combined test set.

deviations whereas score 3 applies to semantic deviations. The results therefore suggest that the GPT-based models rather pick up semantic differences between stimulus and answer sentence compared to the BERT-based model. Looking at the score confusions in Figure 4 shows that the BERT- and rule-based models tend to undervalue answers in this score range, often giving answers with gold score 3 a score of 2 whereas the GPT-based models show the opposite pattern in that answers with gold score 2 often receive a score of 3.

7 Conclusion and Future Work

This paper explored the use of an LLM for scoring a German WEIT. The experiments focused on in-context learning, using k -shot prompting with large k values up to $k = 700$. We considered three sources of variance: The number of examples k , the chosen sample and the non-determinism of LLMs when running the same prompt multiple times. We showed that accuracy varies significantly with k (higher values yielding higher accuracy) and the chosen sample but with sufficiently large k , the effects become non-significant. Furthermore,

while having no significant systematic effect in our regression model, one could see that accuracy is not stable across different runs of the same model. QWK turned out much more robust in this respect, reflecting that the deviations in the scores arising through multiple runs are only small.

Using its best configuration, our GPT k -shot model outperformed a zero-shot GPT model that uses the scoring rubric and no examples. Furthermore, we compared the GPT models to previously obtained results from Chiffigarov et al. (2025), who used a rule-based and a BERT-based model. While the GPT k -shot model could not generally outperform the other models, an interesting pattern emerged when looking at scores 2 and 3, which were previously shown to be the ones that are the most difficult to predict. The GPT k -shot model tends to overestimate responses with score 2 (responses with grammatical errors) while the BERT-based model tends to underestimate responses with score 3 (responses with lexical deviations which are still grammatical). A possible explanation could be that the GPT model relies mostly on semantic similarity between the stimulus and the response sen-

tence, ignoring other kinds of deviations, whereas the BERT-based model pays too little attention to the semantics of a sentence. In future work, we want to investigate this further.

Our study focused on one particular LLM, thus the results may not generalize to other models but may also depend on the architecture, context window etc. Rather, our results show that an LLM does not *necessarily* outperform a rule-based or BERT-based model and that its observed variability, especially when running the same prompt multiple times, may limit its practical applicability. Further investigations are necessary here to find out whether this variability occurs randomly or affects e.g. certain scores in particular.

For future work, we see two promising avenues to improve the automated scoring of a WEIT, firstly experimenting with further prompting strategies for LLMs, especially combining the k -shot approach with the scoring rubric and secondly combining different kinds of models (rule-based, BERT-based and LLMs), exploiting their different strengths and weaknesses across the different scores.

Limitations

The focus of this paper was to study in-depth one prompting strategy (in-context learning with k -shot prompting) and the associated variability when using a particular LLM (GPT-4o) to score a German WEIT. The focus on one LLM limits the generalizability of the results as other models with other architectures and characteristics, e.g. in terms of parameters and context windows, may behave differently. Although our study already captures many factors that influence the performance, there can still be even more sources of variance such as periodic differences in performance of GPT-4o at different points in time (Tschisgale and Wulff, 2026). Furthermore, we did not aim to find the overall best LLM setup for scoring a German WEIT. From our results, one cannot draw the conclusion that k -shot prompting generally works better than rubric-based prompting as we did not systematically tune the rubric-based prompt. Finding the best LLM setup would involve trying different models and more prompting strategies, especially combining the scoring rubric with k -shot prompting, as is state of the art for many applications.

Acknowledgments

The WEIT for German was conceived, developed, and the data collected as part of a research project funded by the Deutsche Forschungsgemeinschaft (DFG), grant number 462766474. We would also like to thank the anonymous reviewers for their very helpful comments.

References

- Berk Atıl, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. 2025. [Non-determinism of “deterministic” LLM system settings in hosted environments](#). In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems*, pages 135–148, Mumbai, India. Association for Computational Linguistics.
- Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. 2025. [In-context learning with long-context models: An in-depth exploration](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12119–12149, Albuquerque, New Mexico. Association for Computational Linguistics.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- Mihail Chiffigarov, Jammila Laâguidi, Max Schellenberg, Alexander Dill, Anna Timukova, Anastasia Drackert, and Ronja Laarmann-Quante. 2025. [Automated scoring of a German written elicited imitation test](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 237–247, Vienna, Austria. Association for Computational Linguistics.
- Dylan G E Gomes. 2022. Should I use fixed effects or random effects when I have fewer than five levels of a grouping factor in a mixed-effects model? *PeerJ*, 10(e12794):e12794.
- C. Ray Graham, Deryle Lonsdale, Casey Kennington, Aaron Johnson, and Jeremiah McGhee. 2008. [Elicited imitation as an oral proficiency measure with ASR scoring](#). In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

- Yue Huang and Joshua Wilson. 2025. [Evaluating LLM-based automated essay scoring: Accuracy, fairness, and validity](#). In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Works in Progress*, pages 71–83, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- Deryle Lonsdale and Carl Christensen. 2011. Automating the scoring of elicited imitation tests. In *Proc. Machine Learning in Speech and Language Processing (MLSLP 2011)*, pages 16–20.
- Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. [Can Large Language Models automatically score proficiency of written essays?](#) In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.
- Benjamin J Millard. 2011. [Oral proficiency assessment of French using an Elicited Imitation Test and Automatic Speech Recognition](#). Master’s thesis, Brigham Young University.
- Shuyin Ouyang, {Jie M.} Zhang, Mark Harman, and Meng Wang. 2025. [An empirical study of the non-determinism of ChatGPT in code generation](#). *ACM Transactions on Software Engineering and Methodology*, 34(2). Publisher Copyright: © 2025 Copyright held by the owner/author(s).
- 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*.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. [Quantifying language models’ sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting](#). In *International Conference on Learning Representations*, volume 2024, pages 25055–25083.
- Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2025. [The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4195–4206, Albuquerque, New Mexico. Association for Computational Linguistics.
- Anna Timukova, Oleksandra Yazdanfar, and Anastasia Drackert. 2026. [So viele Lücken, so wenig Zeit: Die Rolle der Zeit im Konstrukt des deutschen C-Tests anhand der Analyse der Verarbeitungsprozesse](#). *Zeitschrift für Interkulturellen Fremdsprachenunterricht (ZIF)*, 31:81–108.
- Paul Tschisgale and Peter Wulff. 2026. [Daily and weekly periodicity in large language model performance and its implications for research](#). *Preprint*, arXiv:2602.15889.
- Lui Yoshida. 2024. [The impact of example selection in few-shot prompting on automated essay scoring using GPT models](#). In Andrew M. Olney, Irene-Angelica Chounta, Zitao Liu, Olga C. Santos, and Ig Ibert Bittencourt, editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, volume 2150, pages 61–73. Springer Nature Switzerland, Cham. Series Title: Communications in Computer and Information Science.

Kelvi: A Morphological Parser to Support Tamil Literacy

Shankhalika Srikanth¹

shankhalika.srikanth@alumni.utoronto.ca sab.yu@alumni.utoronto.ca

Sabrina Yu¹

Sophia Chan¹

ceasings@gmail.com

Madeline Solis de Ovando¹

madelinesdeo@gmail.com

¹Independent Researcher

Abstract

In this paper we discuss the development of kelvi.ca¹, an open source web-based dictionary and morphological parser designed to aid Tamil learners in developing their literacy skills. Tamil is an agglutinative language and heavily suffixal. Existing Tamil dictionaries only carry stems, not conjugated or inflected forms, and for a beginner learner of the language, isolating the stem in an unfamiliar word can be very challenging. Kelvi provides 1) the stem of any input word alongside its definition, and 2) non-technical descriptions of any suffixes that are part of this input, so that learners will gradually start to recognize these suffixes and be able to understand and produce new Tamil words themselves. In detailing our process of collaborative research, user interviews, suffix database creation, and error analysis, we also hope to show that Kelvi can be adapted for other languages and has the potential to be a useful pedagogical aid for learner literacy development, especially for agglutinative and/or polysynthetic languages which tend to be otherwise underserved in the mainstream.

1 Introduction

Tamil is a Dravidian language spoken in South and Southeast Asia. With its largest speaker population in southern India (73.4 million speakers), it has official language status in the state of Tamil Nadu and the union territory of Puducherry, as well as in Sri Lanka (4.2 million), and Singapore (688.6 thousand). Tamil is also widely spoken in Malaysia (4.8 million) and South Africa (600 thousand). In addition, there are significant diaspora populations in Europe, North America, and Australia; Tamil speakers globally number well over 80

¹Github repositories for the front-end and back-end are accessible at <https://github.com/saltyseadawg/kelvi-frontend> and <https://github.com/saltyseadawg/kelvi-backend> respectively.

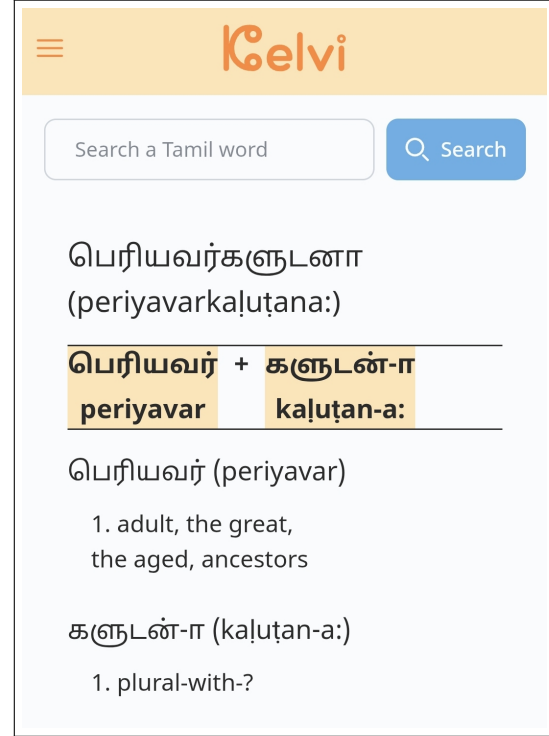


Figure 1: Tamil word as it appears in Kelvi that translates to the question "with the elders/adults/ancestors?"

million (World Population Review, 2026). The diaspora population in Canada largely consists of heritage speakers - that is to say, people who grow up hearing their family speak Tamil, but aren't necessarily entirely fluent themselves (Polinsky and Kagan, 2007). Two of the main complicating features of Tamil literacy are the language's heavily diglossic nature and its agglutinative properties. Tamil is heavily suffixal, with most grammatical information expressed either as nominal or verbal suffixes. See Figure 1, where a complete utterance is expressed as one word.

In written Tamil the higher, formal register is used, where every suffix is phonologically expressed in its entirety. In the colloquial spoken register, however, Tamil words undergo massive phonological reduction, so that many sounds

that are fully expressed in the written form of a word aren't actually pronounced (Annamalai, 2019; Renganathan, 2019). This means that heritage speakers often struggle to recognize the written version of words they have only ever heard in the colloquial spoken register, since the written form “sounds” different from what they're used to (Renganathan, 2019). Conjugated words don't appear in standard dictionaries, only their lemmas do (consider how “go” can be found in an English dictionary but “went” wouldn't be). Learners then face the “dictionary problem”; to summarize Littell et al. (2017), dictionaries are intended to be a tool for language learners, but paradoxically require knowledge of the language to use; in this case, knowledge of what the “dictionary form” or lemma of an unfamiliar word is. To address this issue we built Kelvi, a web-based dictionary and morphological parser that helps a learner who comes across an intimidatingly long Tamil word see how the word is broken down, and what each “part” (morpheme) means in layman terms. The hope is that they come away with a fuller understanding of the word while also starting to recognize Tamil morphological patterns themselves.

2 Literature Review

The one instance known to these authors of a dictionary specifically designed for a heritage/minoritized language that also provides pedagogical information about morphology is the ongoing work on the digital Wendat Dictionary (Lukaniec and Holmes, 2023). More broadly speaking, there are several open-source technologies that have been developed to support language revitalization and maintenance efforts (Brinklow et al., 2020), such as the dictionary-builder Mother Tongues (Littell et al., 2017) and the morphological generator Gramble (Littell et al., 2024) that was used in pedagogical contexts for Oneida (Lu et al., 2024), SENĆOŦEN (Saanich), and nêhiyawêwin (Plains Cree) (Pine et al., 2025a). Effective morphological analysers using FSTs have also been built for similarly agglutinative languages like Turkish (Yıldız et al., 2019). For Tamil specifically there exist tools such as a lemmatizer (Qi et al., 2020), a morphological analyser (Sarveswaran et al., 2021), and a stemmer (Rajalingam, 2025), as well as some pedagogical resources like a dictionary of Tamil verbs (Schiffman et al., 2009) and a few isolated courses/course materials (Annamalai

and Asher, 2002; Prasad, 2025; Cheran, 2001); but to the authors' knowledge there is no other free-to-use pedagogical resource which employs a Tamil morphological analyser.

3 Methods

In accordance with principles behind collaborative community based research (Bucholtz, 2021) and user-centered design methods (Wallisch et al., 2019), we involved target users early on in our explorations of the problem space to ensure we were solving a real issue faced by Tamil learners and providing a solution that learners would want to adopt. We wanted to understand what bottlenecks users were currently facing when searching for Tamil word definitions as well as existing pain points they were experiencing.

3.1 Prototype Design

We built a medium fidelity prototype in Figma, an interface design tool, to test our design (accessible [here](#)). We wanted to know whether (a) users would be able to determine the meaning of a word when presented with the morphological breakdown; (b) users found the morphological breakdown useful; and (c) users would use the tool if it existed.

3.2 Participants

To determine our design requirements, we interviewed 10 Tamil adult speakers over the age of 20 who were asked to self describe their Tamil language abilities in terms of oral fluency and literacy. Of our participants, three were native speakers, six were heritage speakers, and one was a beginner L2 learner with no Tamil background who had only recently begun learning Tamil. Importantly, oral fluency and literacy were not correlated for our speakers. In other words, speakers that were fluent in Tamil were not necessarily strong readers, such as in the case of SM who was a fluent, native speaker of Tamil with weak literacy skills. As shown in Figure 2, most of the participants had stronger oral abilities compared to reading and writing.

3.3 Exploratory Interview

At the beginning of the session, participants were asked to self-describe their Tamil oral and literacy abilities. In addition, since most of the participants were heritage speakers, they were also asked demographic and language related questions to gain insight into their Tamil proficiency. As described by Montrul (2016), background information, such

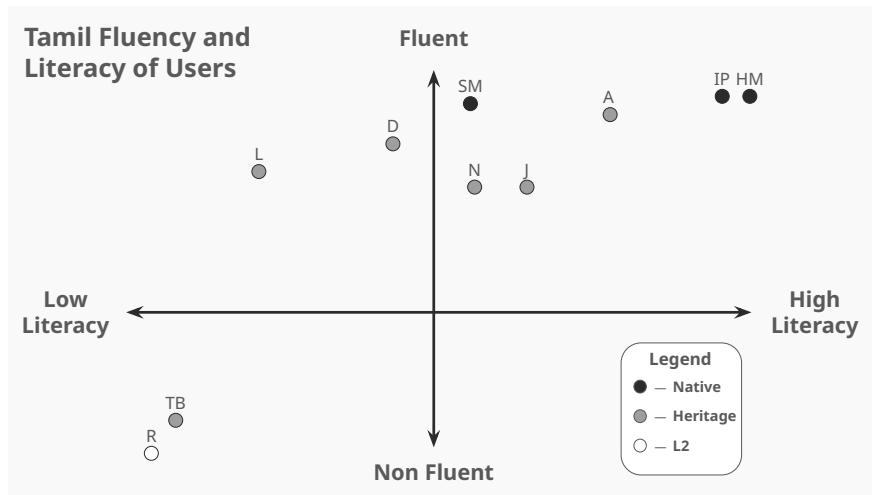


Figure 2: The X-axis represents fluency while the Y-axis represents literacy, forming four quadrants. 6/10 participants fall in the <fluent, high literacy> quadrant, 2/10 participants fall in the <low literacy, non fluent> quadrant, and 2/10 participants fall in the <low literacy, fluent> quadrant. Oral fluency and literacy were thus not correlated among our participants.

as level of education, language of instruction, domains of use of their languages, etc., is important to understanding a heritage speaker’s ability in their heritage language. Although the answers to these questions were mostly used to corroborate the responder’s self description of their Tamil abilities and not used for any formal analyses, the questions themselves have been included in the Appendix for future researchers. Users were then asked to complete four tasks to test the usability of the prototype’s UI as well as input methods, which can also be found in the appendices. After completion of the tasks, users were asked the following questions:

- What do you like/dislike about this tool?
- What features do you wish existed?
- How did each input method compare?
- How did the prototype compare to how you were originally looking up Tamil word definitions in the first task?
- Would you use this tool?

3.4 Interview Results

During the exploratory interviews, participants were asked what tools they were currently using to look up the meaning of Tamil words. The most common answer given was Google Translate, however not all participants were using it just to get the translation of a word. For example, L, who had strong oral abilities but could not read Tamil easily, used Google Translate’s TTS feature to hear

the word and would then be able to figure out what it meant. On the other hand, IP would use Google Translate to get a Tamil word’s romanization which they would then input into the Google search engine to try to find different contexts and uses of the word. For the participants who did rely on the translations provided by Google Translate, several commented that Google’s translations were not always trustworthy. Interestingly, none of our participants mentioned using generative AI tools, however this could be due to the fact that most of our participants were over the age of 25. As Magotra et al. (2016) notes, younger individuals are more likely to adopt new technologies compared to older individuals.

When asked if they would use this tool, seven participants responded with yes, two responded with no, and one with maybe. SM, who is fluent with weak literacy skills, stated they would not use the tool. TB, who is a beginner heritage speaker that cannot read, stated the reason behind their “maybe” response was because they felt that their level of Tamil was too low to use the tool, but would use it if it included more conversational words. IP, who is fluent with high literacy skills, felt the tool was unnecessary and that the interface made it hard to correlate meaning with the word parts.

Conversely, the other participants felt that the tool was quite useful. In N’s case, they thought that learners would find the pedagogical breakdown

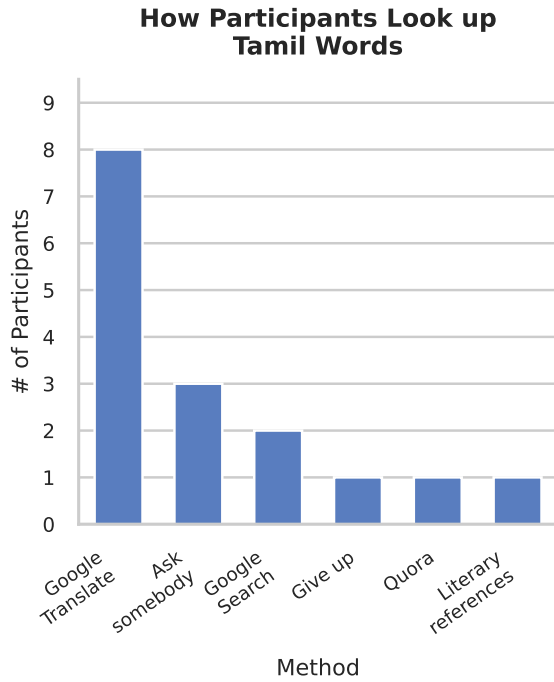


Figure 3: Graph of participants’ methods for looking up Tamil words. Google Translate is the most common at 8 votes, while the five other remaining choices have 3 votes or less.

helpful and wished the tool existed when they were attending Tamil language class during their adolescence. D, who self-described as conversationally fluent, thought that the pedagogical breakdown was not only helpful, but also corrects errors made by Google Translate. Lastly, R, who had only recently started learning Tamil, thought that the definitions provided by the tool were clearer than Google Translate.

4 User Requirements

Based on the results of the exploratory interviews, we established the following user requirements to guide the design of the website.

1. Intuitively understand the morphological breakdown even as an L2 learner.
2. Not overwhelm a beginner learner.
3. Is accessible on the phone.

We also drew inspiration from the Kawennón:nis verb conjugator for Kanyen’kéha (Kazantseva et al., 2018), (which itself was part of the inspiration for Gramble, discussed in Section 5.2), and its aim to assist users in acquiring morphology patterns rather than rote memorization of word definitions.

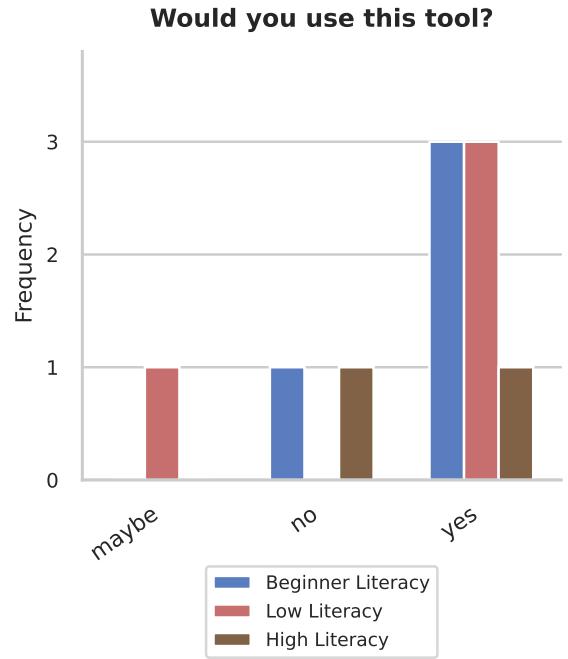


Figure 4: Graph of participants’ responses to if they would use a tool like the prototype. Out of 10 participants, 7 responded ‘yes’, 1 responded ‘maybe’, and 2 responded ‘no’.

5 Systems Design

In this section we discuss the different components of Kelvi and how they interact, as visualized in Figure 5.

5.1 Datasets

We used two open source datasets for the Tamil dictionary component, specifically Tamil Wiktionary data Ylonen (2022) and dictionaries provided by Vasuki (n.d.).

5.2 Suffix Database Generation

The suffix database queried by the system was created using Gramble², a tabular programming language designed for linguistic parsing and generation. We used Gramble to aid us in generating a JSON file³ consisting of Tamil suffixes, their meanings, and other useful (meta)linguistic information that are all stored in Gramble “tiers”. In this section we limit discussion to what tiers (that are then converted into JSON attributes) we created for suffix generation and why.

The primary attributes that every suffix entry in the suffix database have are *text*, *gloss*, *display*, and *add-back*. *text* is a string representing a suf-

²<https://nrc-cnrc.github.io/gramble/>

³File can be found [here](#).

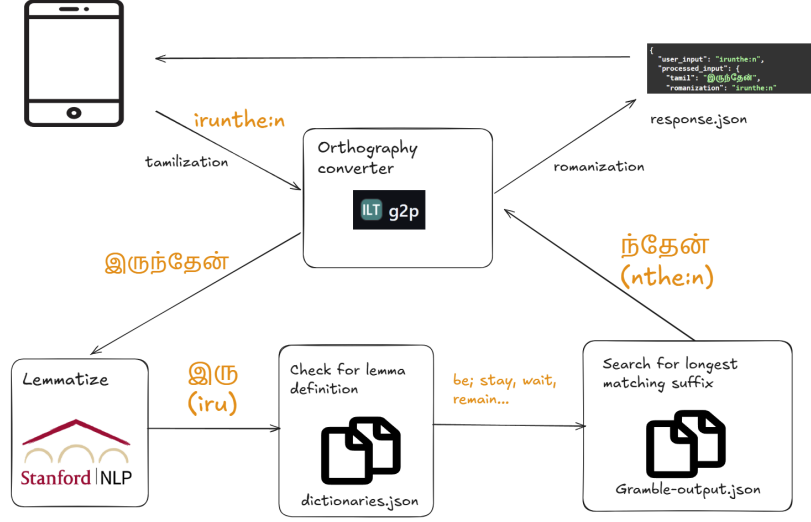


Figure 5: Diagram of Kelvi's backend and frontend.

Plural =		text	gloss	display	add-back
		கள்	[plural]	[கள்]	null
		ங்கள்	[plural]	[கள்]	ம்
		க்கள்	[plural]	[கள்]	null

Figure 6: Excerpt from Gramble file.

fix as it appears in an input word, and is the attribute that the input is matched against. For instance, the input பூதங்கள் *boothangaL* ‘ghosts’ would match to the suffix ங்கள் *ngaL*, as shown in Figure 6. *gloss* is a layman’s definition of this suffix.

display is the version of the matched suffix that is shown to the user: this is generally the same as what is in *text*, but differs in situations where a suffix undergoes minor changes based on phonological environment⁴. For example, the plural suffix கள் *gaL* can also appear as க்கள் *kkaL* or ங்கள் *ngaL* based on the sound it is preceded by, but since கள் is its prototypical form, this is the form of the suffix that is shown (“displayed”) to the user in all cases, as shown in Figure 7. Finally, *add-back* is used as a secondary lemmatizer when the primary fails, to be detailed later in Section 5.5. In this instance, if the lemmatizer were unable to parse பூதங்கள், once the suffix match ங்கள் was identified and removed from the input, the character ம் (*m*) would then be “added back” to this new form to create the valid lemma பூதம் *bootham* (‘ghost’).

⁴Using linguistic terminology, *text* and *display* can roughly be equated to *surface form* and *underlying representation* respectively.

பூதங்கள் (pu:thangka!)		
பூதம் pu:tham	+	கள் ka!
பூதம் (pu:tham) 1. monster, spirit, ghost		
கள் (ka!) 1. plural		

Figure 7: Kelvi UI when given பூதங்கள் *boothangaL* as input.

5.3 Romanized/Tamil Input

Users can input using the Tamil script or romanization (i.e., கேள்வி or *kelvi*). All of the dictionary and morphological data is stored in the Tamil script, so if a user inputs using romanization, their input is first converted into the Tamil script using a custom romanization-to-Tamil mapping that we wrote using G2P, an open-source grapheme-to-

phoneme transducer⁵, (Pine et al., 2022) before being processed. Since there is no universal romanization standard followed by Tamil speakers, our mapping simply attempts to capture the most commonly used spelling patterns. After processing, the output is then returned to the user in both the Tamil script and romanization. The romanization presented here is derived from a Tamil-to-romanization mapping, again written using G2P. The romanization system used here is a combination of commonly used spelling conventions and IPA.

5.4 Lemmatization

Where possible, lemmatization of the Tamil input is performed using Stanza (Qi et al., 2020). The output lemma is then checked against the dictionary data to see if it has an existing dictionary entry. If no valid dictionary entry is found, other strategies are employed to identify the lemma, such as suffix matching.

5.5 Suffix Matching

To identify the potential suffix(es) on an input word, a backwards search is performed on the input to see if there is a match with any suffix included in the Gramble JSON file (generation of this file was discussed in Section 5.2). If there are multiple matches, the longest one (whose length is less than the length of the input) is selected. If a valid lemma was found using the lemmatization process described previously, we return both the lemma and the suffix that we identified. If a valid lemma wasn't found, we create a new potential lemma by removing the text of the matched suffix from the input word. If the potential lemma is now a valid lemma, (i.e., exists as a valid entry in the dictionary data), we return this alongside the suffix as the output to the user. If the potential lemma is still not a valid lemma, we check if the suffix that we found in the Gramble JSON has any material in its *add-back* field: this field contains phonological information that needs to be added to a stem in order to turn it into a lemma. If the *add-back* field is non-empty, we add this material to the potential lemma, confirm if this is now a valid lemma, and return it. If this strategy also fails, we return a "word not found" error to the user.

⁵<https://github.com/NRC-ILT/g2p>

6 Results

6.1 Systems Analysis

Due to limited resources and time (discussed in Section 9), we have not yet been able to perform a formal statistical evaluation. The minimal test set that was used to evaluate performance so far is composed of "edge-case" words, or words we predicted may be tricky to parse based on how our model is constructed. For instance, during early development, we noticed that the Stanford NLP lemmatizer often fails on class III verbs (as defined in Arden, 1910), so we made sure to explicitly compensate for this during the suffix generation process using Gramble, and included several words from this category as part of our testing. Since we observed the remaining six verb classes (Arden, 1910) being accurately lemmatized, fewer of these are represented in our test set as they were assumed to be higher performing. Similarly, there is a higher proportion of words with complex morphophonology and suffixal combinations present in our test set than is necessarily representative of the language, as we predicted these words to be the most challenging to cover. Thus, many "simple" words, for example words of the form [lemma + tense + person-number-gender], are not included in the test set, as once we confirmed that a sample few were working, we assumed others of the type would also be relatively high performing. Of course, formalized random testing is required to verify this. A high level overview of the tool's performance on this test set is shown in Table 1.

Result	# of Tokens	Percentage
Correct	141	50%
Incorrect	46	16%
No result	93	33%
Total	280	

Table 1: Performance of Kelvi on test input words

We primarily tested for presence of lemmas in our dictionary data (shown in Table 2), and accurate lemmatization and suffix identification (shown in Table 3).

Result	# of Tokens	Percentage
Correct	15	75%
Incorrect	1	5%
No result	4	20%
Total	20	

Table 2: Kelvi performance on inputs with no affixes

	# of Tokens	Percentage
Correct	126	48%
Incorrect	45	17%
No result	89	34%
total	260	

Table 3: Kelvi performance on inputs with affixes

For inputs that produced a “word not found” error, we isolated which were failing because the lemma itself was not present in our dictionary data, and which were failing because our tool was not able to lemmatize the word, as shown in Table 4.

Issue	# of Tokens	Percentage
Lemma not in data	10	11%
Bad lemmatization	79	89%
Total	89	

Table 4: Cause for “word not found” errors on inputs with affixes

This delineation will help us isolate which types of structures need to be added to our suffix database. A major category of words that are currently missing is lexical compounds. These are words that have two lemmas, which our model does not currently account for. Some lexemes commonly used in compounds like போது *po:thu* (‘when’, colloquially) can be added to the database as suffixes, but Tamil also has a productive process of forming novel compounds⁶, that will require a more nuanced approach.

For inputs that produced incorrect outputs, we differentiated between whether the error was in the lemma, the suffix, or both (Table 5). The results

Issue	# of Tokens	Percentage
Incorrect lemma	9	20%
Incorrect suffix	32	71%
Both incorrect	4	9%
total	45	

Table 5: Types of errors on inputs with affixes

from Tables 4 and 5 in particular highlight that more work can be done in improving the suffix database, both to increase the accuracy and quantity of suffixes identified, as well as to assist in the lemmatization process. Thus this is where the focus of our future work lies.

⁶For example, the poetic word கண்ணிறங்கா *kaN-NuRanga*: ‘(as) (my) eyes sleep’ is composed of the lemmas கண் *kaN* ‘eye’ and உறங்கு *uRangu* ‘to sleep’, neither of which can be classified as “commonly” used in compounds.

6.2 User Feedback

We conducted a workshop for the launch of the MVP (minimum viable product) for Kelvi to gather user feedback. We also elicited feedback through word-of-mouth, and online through Reddit forums such as r/Tamil and r/LearningTamil.

The feedback received was largely positive. Users requested voice input, romanization input, including example sentences in the output, and providing the definition of the input word in its entirety (not just of the individual morphemes). We were able to add the feature of romanization input following this feedback, and are evaluating the feasibility of incorporating the other requested features; for instance, incorporating TTS (Mahaganapathy and Sarveswaran, 2025).

As discussed further in Section 9, due to lack of time we were not able to conduct a systematic user-feedback study post creation of the MVP. This would certainly ideally be part of a future study. Informally, co-author Srikanth has been able to use the tool as part of her regular Tamil teaching to positive effect. The major drawbacks currently are the low language coverage of the tool, especially when the input is in romanization. However, in the cases where the target words are present in Kelvi, learners found the provided definitions to be useful in understanding both the meaning of a word in its entirety and of its parts.

7 Discussion

Our current bottlenecks and proposed improvements to the software can be summarized as follows:

1. We have limited root-word (i.e., dictionary) data, and results can be improved if we are able to acquire more data.
2. The lemmatizer that we use fails on some inputs; as discussed in (5.5), we are compensating for these gaps using Gramble, which we will continue to do.
3. Some morphemes are homophonous; for example, the suffix -um on a verb means “it will”, but on a noun means “and”. Currently we return both definitions, but if we are able to acquire reliable part-of-speech data, we can refine the output returned to the user.
4. Users are able to input using both the Tamil script and romanization, but the romanization

input is significantly underperforming. This is because there is no romanization standard consistently followed by Tamil speakers, and so the mapping we currently employ cannot account for all possible spellings of a given Tamil word. The romanization practices of Tamil speakers are too irregular to be adequately captured by any one-to-many mapping⁷, no matter how nuanced, and so this feature can only be improved if we are able to train a model on Tamil romanization input and/or implement an approximate search algorithm (Pine et al., 2025b, Littell et al., 2017).

5. Since Kelvi was primarily designed to capture written forms, words in the spoken/informal register are underrepresented. As these words vary widely across a range of dialects, collaboration with speakers from across this span of variation would greatly enhance the coverage of our tool. Currently, Kelvi is only designed to be useful in decoding the meaning of Tamil words inputted in the written/formal register, not for defining words expressed in the spoken/informal register or for converting spoken/informal forms into their written/formal equivalents.

Future work in this area may benefit from the use of machine learning techniques to address some of these challenges. As one example, we can potentially address the variation in romanization input by incorporating transliteration models, such as the one developed by Madhani et al., 2023, to convert between romanization and the Tamil script. Due to time constraints however, we were unable to properly investigate this approach.

8 Conclusion

Although Tamil has a relatively large speaker population, it is still considered a low-resource language which adds a layer of difficulty when developing digital tools and resources for Tamil that

⁷In fact, it is impossible. This is because the letter “n” can be mapped to ன், ன், or ன். ன் is pronounced differently from ன் and ன் and so can be mapped to a distinct romanized character like “N”, but ன் and ன் are pronounced the same. They can’t usefully be mapped to distinct romanized characters because even though they appear in complementary phonological environments 90% of the time, the remaining 10% the distinction is lexically determined, i.e. unpredictable from just the pronunciation, and therefore not something a user who can’t read the Tamil script would be able to know.

make use of data driven innovations in computational linguistics (Tissera and Saadhiq, 2025).

The system described in this paper serves as a starting point for creating pedagogical dictionaries designed for learners of polysynthetic languages and the morphological challenges they commonly face (Kazantseva et al., 2018). Although there is room for improvement in terms of coverage and accuracy, we hope that Kelvi serves as an example of a grassroots initiative that, despite limitations in terms of language data and resources, is able to create a useful language learning tool by taking a user-centered approach and targeting common pain points experienced by learners. The code and data used for the project is open source and will hopefully benefit other individuals and organizations looking to create language learning resources.

9 Limitations

The project received funding (\$10,000 CAD) and had a 1 year deadline as per the agreement for the funding. Unfortunately, the cost of many high quality Tamil dictionary datasets far exceeded the available funding and consequently could not be used. This also impacted the testing process, as all test data had to be created manually and thus was limited to a few hundred inputs. Additionally, this project was done independently and on a voluntary basis without the affiliation of any research institution, which limited the time and resources that could be dedicated to the project.

10 Ethics

We discuss here the ethical considerations of this project on the axes of access, equitable representation of language variation, and data privacy and consent. First, however, we present the positional-ity of the authors.

One co-author (Srikanth) is a heritage Tamil speaker, and through their experience as a learner and teacher of the language, experienced both the challenge of developing their literacy skills and the lack of adequate resources online to help with this. Another co-author (Yu), is a heritage speaker of Cantonese and Mandarin and therefore also familiar with the challenges of heritage language learning. Additionally, both Srikanth and Yu have experience contributing to community-led projects to build digital tools for Indigenous language learning. These professional and lived experiences are

what motivated the authors to create an online resource that could support learners of minoritized and low-resource languages.

10.1 Access

Kelvi was intentionally designed to be open-source and free to use because the motivation behind it was to develop a framework for a language learning tool that could be used by other low resource, minoritized languages. The intention was not to contribute to the history of extraction, exploitation, and gatekeeping of language resources in the field of linguistics (Leonard, 2021) by using language data and putting it behind a pay-wall, especially given that co-author Srikanth is a member of the language community the tool was built for. Open access to language data was deemed appropriate in this case, as Tamil enjoys majority language status in numerous jurisdictions (Tamil Nadu, Sri Lanka, Singapore), and so it wouldn't be considered sensitive information. In future collaborations with other language communities, however, those communities will always have full rights to their data and the choice to keep it open or limited access to only community members, and this is supported by the tool's permissive open source licence.

10.2 Equitable Representation of Language Varieties

We are aware that digital representations of language, especially in educational contexts such as Kelvi, can influence the relative prestige of the varieties used or omitted. Kelvi was deliberately designed to only represent the formal written variety of Tamil. The formal register remains largely uniform across the social axes of religion and region, along which conversely, the spoken register varies greatly. In this way, we hope that Kelvi is accessible to all Tamil users. That being said, we acknowledge that because spoken varieties are not represented in Kelvi, this may contribute to the commonly held belief of language purity that privileges written Tamil over spoken, informal Tamil (Annamalai, 2011). Currently our team is lacking in the linguistic expertise required to adequately represent all of the spoken varieties fairly, and so we seek to form collaborations with speakers of different Tamil varieties to make Kelvi more inclusive in this way.

While formal written Tamil does not undergo significant variation, the same is not true of Tamil written using romanization. There is no "standard"

for romanization, and so each individual uses their own conventions. We don't want to present any one romanization system as a "standard"; in the romanization shown to the user, we use a convention that is associated more with academia, using characters that aren't easily typed (i.e., *t*), to signal to users that the tool isn't meant to influence how they already type. However, the options for romanization input are very limited, and because the romanization is built using a one-to-many mapping (5.3), this mapping does inadvertently end up being prescriptive. Hopefully, as we improve the romanization input, this unintentional side-effect will lessen. Finally, since Kelvi is still only in the MVP stage, it does not have 100% accuracy yet. There are disclaimers on the website informing users of this, so hopefully this lets learners and teachers alike anticipate that there may be the occasional error.

10.3 Data Privacy and Consent

User input is temporarily available for us to view through the web hosting service Kelvi is hosted on. The input is anonymous and can't be tracked to any individuals. We view this input data solely for the purpose of bug-fixing (i.e., keeping track of any failing inputs to add to our testing process).

User feedback from the initial round of user testing has been anonymized, and none of their language data was used in the building of Kelvi or accessible to anyone outside of the team. The interviews were recorded with their consent and they were informed that the recordings would only be consulted for the purposes of improving the tool, and would not be shared with anyone else.

Additional feedback solicited once the MVP was built was encouraged to be submitted through the anonymized feedback forms on the Kelvi website. Some individuals contacted the authors directly; regardless, all feedback is kept anonymized here and in any future publications.

11 Acknowledgements

This project was made possible with support from the Audrey Wearn Language Prize and the Future Design School. We would also like to thank Fizza Ahmed, Vasumathi Srikanth, Srikanth Mangalam, Aidan Pine, Patrick Littell, Sowmya Vajjala, Roland Kuhn, and the Indigenous Languages Technology project team at the National Research Council of Canada. Finally, thank you to the anonymous reviewers for their feedback.

References

- E. Annamalai. 2011. *Social dimensions of modern Tamil*, 1st edition. Cre-A:, Chennai.
- E. Annamalai. 2019. Modern Tamil. In Sanford B. Steever, editor, *The Dravidian languages*, 2 edition. Routledge. Num Pages: 27.
- E. Annamalai and R. E. Asher. 2002. *Colloquial Tamil: The Complete Course for Beginners*. Taylor & Francis Group, London, UNITED KINGDOM.
- Albert Henry Arden. 1910. *A progressive grammar of common Tamil*. Madras : Society for Promoting Christian Knowledge.
- Nathan Thanyehténhas Brinklow, Patrick Littell, Delaney Lothian, Aidan Pine, and Heather Souter. 2020. Indigenous Language Technologies & Language Reclamation in Canada. In *International Conference Language Technologies for All (LT4All): Enabling Linguistic Diversity and Multilingualism Worldwide*. UNESCO.
- Mary Bucholtz. 2021. *Community-Centered Collaboration in Applied Linguistics*. *Applied Linguistics*, 42(6):1153–1161.
- Elango Cheran. 2001. *Learn Tamil*.
- Anna Kazantseva, Aidan Pine, Owennatekha Brian Maracle, and Ronkwe'tiyohstha Joe Maracle. 2018. *Kawennón:nis: the Wordmaker for Kanyen'kéha (Ohsweken Mohawk)*. In *COLING: The 27th International Conference on Computational Linguistics*.
- Wesley Y. Leonard. 2021. *Toward an Anti-Racist Linguistic Anthropology: An Indigenous Response to White Supremacy*. *Journal of Linguistic Anthropology*, 31(2):218–237.
- Patrick Littell, Aidan Pine, and Henry Davis. 2017. *Waldayu and Waldayu Mobile: Modern digital dictionary interfaces for endangered languages*. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 141–150, Honolulu. Association for Computational Linguistics.
- Patrick Littell, Darlene Stewart, Fineen Davis, Aidan Pine, and Roland Kuhn. 2024. Gramble: A Tabular Programming Language for Collaborative Linguistic Modeling. In *LREC-COLING 2024*. ACL.
- Yanfei Lu, Patrick Littell, and Keren Rice. 2024. *Empowering Oneida language revitalization: Development of an Oneida verb conjugator*. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5757–5767, Torino, Italia. ELRA and ICCL.
- Megan Lukaniec and Martin Holmes. 2023. *Modern Wendat Lexicography: Using XML to Reflect the Grammar and Lexicon of an Iroquoian Language*. *Dictionaries: Journal of the Dictionary Society of North America*, 44(2):75–105.
- Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Khapra. 2023. *Aksharantar: Open Indic-language transliteration datasets and models for the next billion users*. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 40–57, Singapore. Association for Computational Linguistics.
- Irbha Magotra, Jyoti Sharma, and Supran Kumar Sharma. 2016. *Assessing personal disposition of individuals towards technology adoption*. *Future Business Journal*, 2(1):81–101.
- Ahrane Mahaganapathy and Kengatharaiyer Sarveswaran. 2025. *A survey and evaluation of text-to-speech systems for the tamil language*. *Natural Language Processing Journal*, 12:100171.
- Silvina Montrul. 2016. Heritage languages and heritage speakers. In *The Acquisition of Heritage Languages*, pages 13–40. Cambridge University Press, Cambridge.
- Aidan Pine, Erica Cooper, David Guzmán, Eric Joanis, Anna Kazantseva, Ross Krekoski, Roland Kuhn, Samuel Larkin, Patrick Littell, Delaney Lothian, Akwiratékhá' Martin, Korin Richmond, Marc Tessier, Cassia Valentini-Botinhao, Dan Wells, and Junichi Yamagishi. 2025a. *Speech generation for indigenous language education*. volume 90, page 101723.
- Aidan Pine, David Huggins-Daines, Carmen Leeming, Patrick Littell, Timothy Montler, Heather Souter, and Mark Turin. 2025b. *Zero-shot query generation for approximate search algorithm evaluation*. In *Proceedings of the Eight Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 65–73, Honolulu, Hawaii, USA. Association for Computational Linguistics.
- Aidan Pine, Patrick Littell, Eric Joanis, David Huggins-Daines, Christopher Cox, Fineen Davis, Eddie Antonio Santos, Shankhalika Srikanth, Delasie Torkornoo, and Sabrina Yu. 2022. Gi2Pi: Rule-based, index-preserving grapheme-to-phoneme transformations. In *5th Workshop on the Use of Computational Methods in the Study of Endangered Languages*. ComputEl-5.
- Maria Polinsky and Olga Kagan. 2007. *Heritage Languages: In the 'Wild' and in the Classroom*. *Language and Linguistics Compass*, 1(5):368–395.
- Arun Prasad. 2025. *Tamil for Beginners*.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. *Stanza: A Python natural language processing toolkit for many human languages*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
- Damodharan Rajalingam. 2025. *rdamodharan/tamil-stemmer*. Original-date: 2010-12-27T17:37:49Z.

Vasu Renganathan. 2019. Heritage language environment (HLE) and its implications upon language curriculum: a case of Tamil heritage students and designing a differentiated curriculum. In *Langues de l'Inde en diasporas : maintiens et transmissions*, Paris.

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. *ThamizhiMorph: A morphological parser for the Tamil language*. *Machine Translation*, 35(1):37–70.

Harold F. Schiffman, Vasu Renganathan, Prathima Christdas, S. Palanisamy, Thanam Baba, K. Karunakaran, Ganesh Krishnamurti, C.M. Pradeep, and S. Radhakrishnan, editors. 2009. *An English dictionary of the Tamil verb*. South Asia Studies Dept. University of Pennsylvania, Philadelphia, Pa.

Muditha Tissera and Hassaan Saadhiq. 2025. *Natural Language Processing Resources for Tamil Language: A Systematic Review*. *Journal of Information and Communication Convergence Engineering*, 23(4):236–258.

Vishas Vasuki. n.d. *Dharma Via Vishvas*.

Anne Wallisch, Olga Sankowski, Dieter Krause, and Kristin Paetzold. 2019. *Overcoming fuzzy design practice: revealing potentials of user-centered design research and methodological concepts related to user involvement*. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–9.

World Population Review. 2026. *Official use of Tamil by country 2026*.

Olca Taner Yıldız, Begüm Avar, and Gökhan Ercan. 2019. *An open, extendible, and fast Turkish morphological analyzer*. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1364–1372, Varna, Bulgaria. INCOMA Ltd.

Tatu Ylonen. 2022. *Wiktextextract: Wiktionary as Machine-Readable Structured Data*. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.

A Interview Questions

Sociolinguistic questions used for the exploratory interview:

- How old are you?
- How would you describe your fluency in Tamil? What situations do you tend to use Tamil in? Speaking? Reading? Writing?
- Where did you go to school?

The following tasks were given to the participants to complete using the prototype. Note that the Tamil words were presented in text; no romanization was given and facilitators did not read out the words.

1. Find the meaning of பெரியவர்களுக்கு using the prototype interface. Try entering the word using Tanglish⁸ / English letters.
2. Find the meaning of அறிந்துகொண்டேன் but type the word in Tamil.
3. Find the meaning of பெருகிவிட்டால் but use your voice.
4. Find the meaning of சிதறிவிடும் using your preferred input method from the previous tasks.

⁸Colloquial term used to refer to the code-mixing of Tamil and English in casual speech.

From Questions to Assessment Tuples: A Multi-Agent Framework with Bloom-Specialized Agents and Automated Verification

Gee-Lyle Wong Runcong Zhao Yulan He Jiazheng Li

King's College London

{gee-lyle.wong, runcong.zhao, yulan.he, jiazheng.li}@kcl.ac.uk

Abstract

Automatic question generation with large language models has advanced rapidly, yet producing *assessment-ready* items, complete with mark schemes and expected answers, remains challenging, especially when generation must reliably target higher-order cognitive levels in Bloom's Taxonomy. We propose a multi-agent, multi-stage framework that generates structured assessment tuples for both short-answer questions (SAQs) and scenario-based questions (SBQs), combining Bloom-specialized generation agents with staged decomposition and automated verification. We further introduce a rubric-guided LLM-as-a-judge evaluation framework with Bloom-specific alignment metrics. Experiments on university-level AI course material across five generation pipelines show that prompt-level Bloom conditioning alone is insufficient to reliably achieve cognitive control. In contrast, our structured approach yields consistent and notable improvements in alignment, mark scheme quality, and output yield, particularly for higher-order Bloom levels over baseline pipelines.

1 Introduction

Assessment is central to teaching and learning, supporting formative practice, diagnostic feedback, and summative evaluation. However, producing high-quality assessment items is time-intensive and requires both domain expertise and pedagogical design, limiting scalability (Kurdi et al., 2020; Das et al., 2021). Automatic Question Generation (AQG) and Automatic Item Generation (AIG) have therefore emerged as key research directions, evolving from rule-based systems to neural and, more recently, large language model (LLM)-based approaches (Zhang et al., 2021; Kurdi et al., 2020; Tan et al., 2025).

Despite this progress, most work focuses on generating question text alone or on multiple-choice questions (MCQs), leaving open-ended assessment types such as short-answer questions (SAQs) and

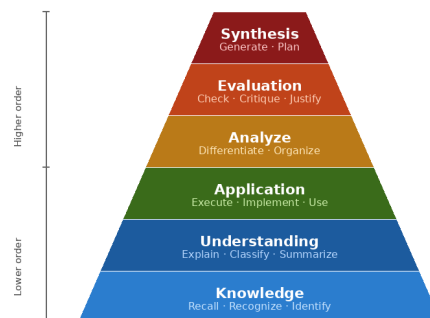


Figure 1: Bloom's Taxonomy of Educational Objectives (LW et al., 2001), comprising six cognitive levels ordered by increasing complexity. Reliably generating assessment items at higher-order levels is a central challenge addressed in this work.

scenario-based questions (SBQs) comparatively underexplored (Song et al., 2025; Chan et al., 2025). SAQs require brief, constructed responses that demonstrate conceptual understanding, while SBQs present realistic, course-grounded scenarios that require students to apply their knowledge and reasoning using both the provided context and learned material.

Open-ended formats play a critical role in assessing deeper understanding and reasoning, beyond the constrained nature of MCQs (Kurdi et al., 2020; Das et al., 2021; Song et al., 2025). However, their effectiveness depends on more than the question alone. Reliable assessment requires aligned marking schemes and exemplar answers, introducing additional complexity: systems must generate coherent, pedagogically grounded assessment items with clearly scoped tasks, objective grading criteria, and internal consistency across components. This shifts the problem from question generation to *structured assessment generation*, where multiple components must jointly satisfy pedagogical constraints.

A second challenge lies in *cognitive control*. Bloom's Taxonomy is widely used to define learning objectives, and prior work has explored Bloom-

aware prompting to control question difficulty. While such approaches improve over unconstrained generation, they remain unreliable, particularly at higher-order cognitive levels such as *analyze*, *evaluate*, and *synthesis* (Scaria et al., 2024; Maity et al., 2024; Yaacoub et al., 2025). More broadly, aligning generated items with intended cognitive demand remains an open problem (Uto et al., 2023).

A third limitation concerns *evaluation*. Question generation is inherently one-to-many, making reference-based metrics, such as lexical overlap, unreliable proxies for quality (Oh et al., 2023; Nguyen et al., 2024). Existing evaluation approaches, ranging from reference-free metrics to LLM-as-a-judge, remain fragmented and fail to capture whether generated items are coherent, gradable, and cognitively aligned as complete assessment units (Mohammadshahi et al., 2023; Liu et al., 2023; Hashemi et al., 2024).

To address these challenges, we propose a unified, Bloom-controlled framework for generating assessment-ready items from course material. Generation is decomposed into a structured multi-stage process that produces complete assessment tuples: for SAQs, a question, mark scheme, and expected answer; for SBQs, an additional scenario is constructed and optionally refined. Bloom-specialized generation enables explicit cognitive control, while automated verification enforces grounding, alignment, and internal consistency. We further introduce a rubric-guided LLM-as-a-judge evaluation framework tailored to structured assessment generation. The framework separates stem-level quality from full assessment consistency, enabling fine-grained analysis of failure modes. We also define Bloom-specific metrics that quantify alignment across all cognitive levels and within higher-order levels.

We evaluate five pipelines spanning zero-shot, Bloom-aware, and multi-stage approaches across both SAQs and SBQs using Artificial Intelligence course material. Results show that prompt-level Bloom conditioning alone does not achieve consistent cognitive targeting in our setting, while staged generation with Bloom specialization and verification yields improvements in alignment, structural quality, and output yield, particularly at higher-order levels.

Our contributions are as follows:

- We introduce a Bloom-specialized, multi-agent, multi-stage framework for both SAQs and SBQs, including scenario construction and refinement for contextualized reasoning.

- We develop a multi-criteria LLM-as-a-judge framework that evaluates pedagogical quality, structural validity, and Bloom alignment.
- We conduct an empirical comparison of five generation pipelines across two assessment formats, showing that staged Bloom specialization with verification improves alignment, structural quality, and yield over prompt-level conditioning.

2 Related Work

Automatic question and item generation. Recent surveys highlight the flexibility of LLMs for generating assessment items across domains, but also emphasize limitations in quality control, pedagogical grounding, and evaluation practices (Kurdi et al., 2020; Zhang et al., 2021; Das et al., 2021; Tan et al., 2025; Song et al., 2025). Much of this work focuses on multiple-choice questions, while open-ended formats such as SAQs and SBQs remain comparatively underexplored despite their importance for assessing higher-order understanding (Song et al., 2025).

Assessment consistency and explainable scoring. Related work on automated student answer scoring and feedback generation shows the importance of aligning outputs with rubrics, key answer elements, rationales, and curriculum context. Prior systems use ChatGPT-generated rationales, thought-tree decomposition, preference calibration, and educator-in-the-loop workflows to improve explainability and assessment reliability (Li et al., 2023, 2024b,a; Zhao et al., 2025). These works motivate our focus on generating assessment-ready tuples rather than question stems alone.

Cognitive control and Bloom’s taxonomy. Controlling cognitive complexity is a central goal in educational generation. Bloom-aware prompting can improve alignment compared to standard prompting, but performance remains inconsistent, particularly for higher-order categories (Scaria et al., 2024; Maity et al., 2024; Yaacoub et al., 2025; Uto et al., 2023; Yadav et al., 2025). Our work therefore treats Bloom control as part of the generation pipeline, rather than relying solely on prompt-level conditioning.

Evaluation of question generation. Evaluation remains challenging because question generation is inherently one-to-many. Metrics that compare generated questions against fixed reference questions, such as BLEU- or ROUGE-style overlap metrics, correlate poorly with human judgment and may penalize valid alternative questions, motivating reference-free, multi-dimensional, and LLM-

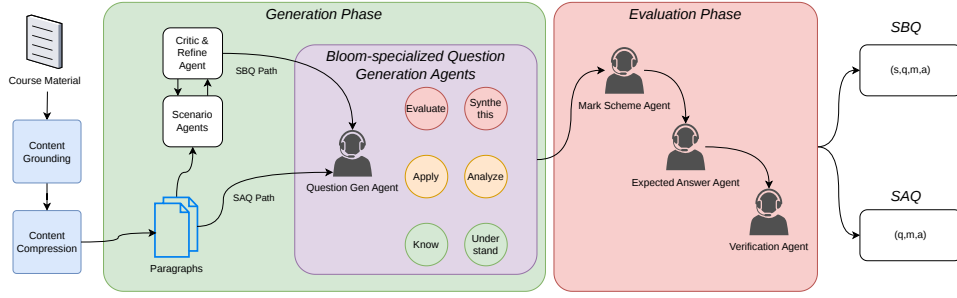


Figure 2: Overview of the proposed Bloom-guided multi-agent generation framework. The pipeline begins with course grounding and relevance filtering, followed by course-content compression. For SBQs, a dedicated scenario generation stage is applied and refined through critique before question generation; for SAQs, question generation proceeds directly from the filtered and compressed course content. Both paths then generate a mark scheme and expected answer, after which a verification agent checks Bloom alignment, grounding, and structural validity. Final outputs are tuples (q, m, a) for SAQs and (s, q, m, a) for SBQs.

based alternatives (Oh et al., 2023; Nguyen et al., 2024; Mohammadshahi et al., 2023; Fu et al., 2024; Liu et al., 2023; Hashemi et al., 2024). However, some methods often focus on question-level quality rather than full assessment consistency, so our evaluation combines stem- and scenario-level criteria with triple-level checks across the question, mark scheme, and expected answer, alongside Bloom-specific alignment metrics.

Structured and multi-stage generation. Prior work has decomposed educational assessment into multiple components, including question, answer, and distractor generation in MCQ pipelines, or key-element matching, rubric application, rationale generation, and refinement in explainable scoring systems (Li et al., 2024b, 2025). Iterative refinement and self-feedback further motivate the use of critique and verifier models for improving generated outputs (Madaan et al., 2023; Li et al., 2025). Our work extends this structured view to open-ended assessment generation by constructing full assessment tuples through multi-stage generation with explicit verification.

Limitations of existing approaches. Across these directions, three limitations persist: (i) *limited support for assessment-ready outputs beyond question text*, (ii) *unreliable control over cognitive complexity, particularly at higher-order levels*, and (iii) *fragmented evaluation practices that do not capture full assessment quality*. Our work addresses these gaps through a unified framework that integrates structured generation, Bloom-specialized control, and rubric-based evaluation for complete assessment items.

3 Methodology

We propose a unified, Bloom-controlled generation framework for producing assessment-ready

items from course documents. Rather than generating questions in isolation, the framework produces structured assessment tuples, enabling both cognitive control during generation and structural validation after generation. It supports two assessment variants within the same system: short-answer questions (SAQs), generated directly from source content, and scenario-based questions (SBQs), which introduce an intermediate scenario representation prior to question generation.

Formally, let D denote an extracted course document, and let $b \in \mathcal{B}$ denote a target Bloom level (knowledge, comprehension, application, analysis, evaluation, synthesis)¹. The framework generates an assessment output y , defined as $y = (q, m, a)$ for SAQs and $y = (s, q, m, a)$ for SBQs, where s is a scenario, q is the question, m is the mark scheme, and a is an expected full-mark answer.

The overall system can be viewed as a controlled generation function

$$\mathcal{F}(D, b, t) \rightarrow y,$$

where $t \in \{\text{SAQ}, \text{SBQ}\}$ specifies the assessment type.

3.1 Generation Framework

The framework is formulated as a multi-agent, multi-stage generation process. Rather than relying on a single prompt to jointly produce all outputs, generation is decomposed into specialized components that each operate on a narrower subtask. This decomposition serves three purposes: it improves control over Bloom level, reduces interference between subtasks, and enables explicit verification of intermediate and final outputs.

¹We follow the original Bloom’s Taxonomy (LW et al., 2001) with six cognitive levels ordered by increasing complexity.

At a high level, the framework consists of four functional stages:

1. **Content grounding:** identify the source content from which an item should be generated.
2. **Question construction:** generate either a direct question (SAQ) or a scenario followed by a question (SBQ), conditioned on a target Bloom level.
3. **Assessment completion:** generate a mark scheme and then an expected answer conditioned on the previously generated outputs.
4. **Verification and filtering:** apply automated checks to enforce Bloom alignment, grounding, and internal consistency.

This design treats SAQs and SBQs as two paths through the same framework. Both share the same downstream structure (question $\rightarrow$ mark scheme $\rightarrow$ expected answer), while SBQs introduce an additional scenario construction and refinement step before question generation.

3.2 Framework Components

Bloom-specialized question agents. To improve cognitive control, the framework uses Bloom-specialized generation instructions. For each target level b , a dedicated question-generation agent is configured with level-specific guidance about the expected cognitive operation, response format, and acceptable task type. This replaces a single generic prompting strategy with a family of specialized generators $G_q^{(b)}(\cdot)$ for $b \in \mathcal{B}$.

For SAQs, questions are generated directly from source content as $q = G_q^{(b)}(D)$, while for SBQs, question generation is conditioned on a refined scenario, $q = G_q^{(b)}(s')$. This per-level specialization is central to the framework, as the objective is not only to generate valid questions, but to control their cognitive demand.

Scenario construction agent (SBQ variant). SBQs introduce an intermediate contextual representation. A scenario agent generates a course-grounded situation $s = G_s(D)$, intended to provide a context in which subsequent reasoning is situated. Rather than adding superficial realism, this stage transforms source knowledge into a setting that supports application-, analysis-, or evaluation-oriented assessment.

Because unconstrained scenario generation may introduce answer leakage or implicit guidance, the generated scenario is not used directly and is instead passed to a refinement stage.

Scenario critique-refinement agent (SBQ vari-

ant). The framework incorporates a constraint-aware critique-refinement stage. A critique function $V_s(\cdot)$ evaluates the generated scenario with respect to the source content, producing a validation signal $r_s = V_s(D, s)$ that reflects compliance with pedagogical constraints such as neutrality, absence of answer leakage, and suitability for downstream questioning. If violations are detected, the scenario is revised as $s' = R_s(D, s, r_s)$; otherwise, $s' = s$. This ensures that the scenario remains a neutral, information-bearing context rather than implicitly encoding the answer.

Mark scheme generation agent. Given a generated question q and context c (where $c = D$ for SAQs and $c = s'$ for SBQs), the framework generates a structured mark scheme $m = G_m(c, q)$. The mark scheme is represented as a set of atomic marking points with associated integer marks, enabling transparent grading and facilitating downstream verification.

Expected answer generation agent. The final generation stage produces an exemplary full-mark answer $a = G_a(c, q, m)$ conditioned on the context, question, and mark scheme. Conditioning on the mark scheme ensures that the generated answer aligns with the grading criteria rather than being only superficially plausible.

Verification agent. All candidate outputs are passed through a final verification layer before being retained. The verifier evaluates whether generated components satisfy structural and pedagogical constraints, formalized as

$$\phi(y; D, b, t) \in \{0, 1\}.$$

Only outputs with $\phi = 1$ are retained. This stage functions as a quality-control mechanism and is a core component of the framework rather than a post-hoc filtering step.

3.3 SAQ and SBQ Instantiations

Although the framework is unified, the two assessment formats differ in how the question context is constructed.

SAQ instantiation. For SAQs, question generation is performed directly from course content. Formally, we define $q = G_q^{(b)}(D)$, $m = G_m(D, q)$, and $a = G_a(D, q, m)$. The candidate tuple $y = (q, m, a)$ is then passed through verification:

$$\mathcal{F}_{\text{SAQ}}(D, b) = y \cdot \mathbb{I}[\phi(y; D, b) = 1],$$

where only tuples satisfying the verification predicate ϕ are retained as valid outputs. This variant is

suited to direct recall, explanation, application, or short analytical reasoning grounded in the provided course material.

SBQ instantiation. For SBQs, the framework first constructs and conditionally refines a scenario before question generation. Specifically, we define $s = G_s(D)$ and $r_s = V_s(D, s)$, where $V_s(\cdot)$ evaluates the scenario against pedagogical constraints. The refined scenario is then given by

$$s' = \begin{cases} R_s(D, s, r_s) & \text{if } r_s \text{ indicates a violation,} \\ s & \text{otherwise.} \end{cases}$$

Question generation then proceeds as $q = G_q^{(b)}(s')$, followed by $m = G_m(s', q)$ and $a = G_a(s', q, m)$. The candidate tuple $y = (s', q, m, a)$ is then verified:

$$\mathcal{F}_{\text{SBQ}}(D, b) = y \cdot \mathbb{I}[\phi(y; D, b) = 1].$$

The scenario introduces a context in which students must interpret information before answering, enabling contextualized application and higher-order reasoning while preserving the same downstream assessment structure as SAQs.

3.4 Control and Verification Logic

The framework combines *generation control* and *verification*. Generation control is achieved through Bloom-specialized agents and staged conditioning, while verification acts as a filtering mechanism that enforces structural and pedagogical validity of the generated outputs.

Given a candidate output y , the verification stage evaluates whether it satisfies a set of core constraints grounded in the source content and target Bloom level. These checks cover four aspects: (i) **Bloom alignment**, ensuring the question matches the intended cognitive level; (ii) **grounding**, requiring all generated components to be supported by their source context (D for SAQs, and both D and s' for SBQs); (iii) **mark scheme quality**, enforcing completeness, non-redundancy, and alignment with the question; and (iv) **answer fidelity**, ensuring the expected answer satisfies all marking criteria without introducing unsupported information.

Only candidates that satisfy all constraints are retained. The framework therefore operates not as a single generator, but as a controlled generation system in which outputs are iteratively produced and selectively accepted based on explicit validation criteria.

3.5 Discussion of Design Rationale

The framework is designed around the observation that assessment generation is inherently compositional. A high-quality item is not only a well-phrased question; it is a coherent structure linking source content, cognitive demand, grading criteria, and an exemplary answer. Treating these components as jointly generated in one step makes control difficult and validation opaque. By decomposing the task into specialised agents with explicit interfaces, the proposed framework improves controllability, makes failures more localisable, and supports direct comparison between simpler and more structured generation strategies in the experimental section. Full prompt templates, per-stage implementation details, Bloom-level generation guidelines, and the generation-time verification predicate are provided in Appendix A.

4 Experiments and Results

4.1 Experiment Setup

We evaluate automatic question generation on university level *Artificial Intelligence* course materials derived from publicly available sources. The source documents are preprocessed into structured content representations, from which assessment items are generated.

Experiments are conducted across two assessment formats: short-answer questions (SAQs) and scenario-based questions (SBQs). While both formats aim to assess conceptual understanding, SBQs additionally require the generation of a contextual scenario that conditions the question, enabling applied and higher-order reasoning.

Generation is performed using **GPT-4.1**, **GPT-4o-mini**, and **Gemini-2.5 Flash** (OpenAI et al., 2024; Comanici et al., 2025). Results are reported for each pipeline-generator pair (p, g) to isolate the effect of both prompting strategy and underlying model capability. Evaluation is performed using three LLM-as-a-judge models: **GPT-4o-mini**, **GPT-4.1-mini**, and **Gemini-2.5 Flash**. Unless otherwise stated, reported criterion and Bloom scores are averaged across these evaluator models.

4.2 Baselines and Proposed Pipelines

We evaluate five generation pipelines designed to progressively address limitations in controllability, decomposition, and cognitive alignment.

Zero-shot (ZS). A single prompt generates the full assessment item in one step. For SAQs, this corresponds to generating a (question, mark scheme,

expected answer) triple, while for SBQs the scenario is generated jointly with the question.

Zero-shot + Bloom (ZS+B). Extends zero-shot prompting by explicitly conditioning on a target Bloom level, testing whether cognitive alignment can be achieved through prompt-level control alone.

Multi-stage zero-shot (MZS). Decomposes generation into sequential stages (question $\rightarrow$ mark scheme $\rightarrow$ answer), reducing task interference. For SBQs, scenario and question generation are performed jointly in the first stage.

Multi-stage zero-shot + Bloom (MZS+B). Integrates Bloom-level conditioning into the multi-stage pipeline during question generation, assessing whether decomposition and cognitive control jointly improve alignment.

Proposed pipeline. Our approach extends the multi-stage Bloom-aware pipeline in three ways. First, question generation uses *Bloom-specialized agents* with per-level system instructions rather than a single generic Bloom prompt. Second, for SBQs, a dedicated scenario generation stage is followed by critique and refinement to enforce grounding, neutrality, and the absence of answer leakage. Third, an explicit verification layer filters candidates based on Bloom alignment, grounding, and internal consistency.

4.3 Evaluation Criteria and Metrics

We define a structured set of criteria capturing both *question quality* and *assessment consistency*, evaluated at two levels: the question stem (or scenario and question for SBQs) and the full assessment triple. All criteria are evaluated independently using rubric-guided LLM judges to reduce cross-criterion interference. Scores are aggregated across three evaluator models to test whether the main conclusions are robust across different evaluators. We use the following abbreviations throughout the paper.

Stem-level criteria. **CG** (course grounding): whether the question is derived from source material. **Cla.** (clarity): linguistic precision and absence of ambiguity. **STI** (single-task integrity): coherence of the assessed objective. **OG-S** (objective gradability, stem): whether the question admits a clearly gradable answer. For SBQs, we additionally evaluate **SRN** (scenario relevance/necessity) and **SG** (scenario grounding).

Triple-level criteria. **OG-T** (objective gradability, triple-aware): whether the full triple supports consistent and objective grading. **MSQ** (mark scheme

quality): coverage, structure, and non-redundancy of marking points. **AF** (answer fidelity): alignment between the expected answer and the mark scheme.

Bloom-specific metrics. **BLA** (Bloom level alignment): a soft alignment score accounting for the hierarchical structure of Bloom’s Taxonomy, defined as $BLA_i = 1 - |\text{gap}(b_i, \hat{b}_i)|/L$, where b_i is the target level, $\hat{b}_i$ the inferred level, and L the maximum possible gap. **BA** (Bloom accuracy): exact-match alignment between intended and inferred Bloom levels. **HOSR** (higher-order success rate): exact-match alignment restricted to higher-order levels ($\{\text{analysis, evaluation, synthesis}\}$). Full formal definitions are provided in Appendix B.

4.4 Evaluation Protocol

We employ three complementary evaluation paradigms: criterion-based scoring, matched-task pairwise comparison, and inter-judge agreement analysis.

Criterion-based evaluation. Each generated item receives a discrete score $s \in \{0, 1, 2\}$ per criterion from each evaluator model, normalized to $[0, 1]$ and averaged across judges. We aggregate using discrimination-weighted scoring, where criteria that better separate pipeline performance contribute proportionally more to the final composite score. Full formalization is given in Appendix B.

Matched-task pairwise comparison. To compare pipelines directly, we match them on task slots defined by source file, target Bloom level, and generation model. For each matched triple, we compare representative outputs in two modes: *raw* (first generated candidate, measuring intrinsic quality) and *production* (highest-scoring candidate, measuring end-to-end pipeline quality after selection). Outcomes are aggregated into win, tie, and loss rates with bootstrap 95% confidence intervals.

Inter-judge agreement. To examine whether the evaluation is stable across LLM judges, we compute agreement statistics over GPT-4o-mini, GPT-4.1-mini, and Gemini-2.5 Flash. For rubric criteria, we report exact three-judge agreement, the mean standard deviation of normalized scores, and mean pairwise Pearson correlation. For inferred Bloom level, we additionally report exact three-judge agreement and mean pairwise agreement over predicted Bloom labels.

4.5 Results

Overview Tables 1–5 summarize criterion-level performance, Bloom metrics, inter-judge agreement, and matched-task pairwise comparisons

		SAQ							SBQ									
Pipe.	Generator	OG-T	OG-S	MSQ	AF	STI	Cla.	CG	OG-T	OG-S	SRN	SG	MSQ	AF	STI	Cla.	CG	Avg
ZS	GPT-4.1	0.931	0.896	0.519	0.959	0.918	0.991	0.981	0.927	0.827	0.555	0.845	0.515	0.976	0.858	0.903	0.924	0.845
	GPT-4o-mini	0.959	0.921	0.447	0.959	0.921	0.991	0.981	0.852	0.826	0.595	0.894	0.549	0.962	0.845	0.909	0.955	0.848
	Gemini-2.5 Flash	0.961	0.936	0.426	0.980	0.917	0.995	0.946	0.919	0.820	0.599	0.829	0.653	0.964	0.833	0.887	0.964	0.852
ZS+B	GPT-4.1	0.950	0.925	0.431	0.965	0.918	0.981	0.959	0.904	0.825	0.610	0.842	0.523	0.980	0.879	0.873	0.929	0.843
	GPT-4o-mini	0.943	0.912	0.236	0.953	0.918	0.991	0.975	0.946	0.833	0.588	0.858	0.603	0.956	0.912	0.956	0.931	0.844
	Gemini-2.5 Flash	0.977	0.921	0.398	0.944	0.921	1.000	0.986	0.880	0.870	0.681	0.787	0.662	0.968	0.856	0.856	0.954	0.854
MZS	GPT-4.1	0.863	0.877	0.352	0.944	0.896	0.986	0.957	0.888	0.818	0.551	0.827	0.554	0.990	0.856	0.876	0.939	0.823
	GPT-4o-mini	0.867	0.892	0.296	0.963	0.885	0.981	0.975	0.875	0.808	0.609	0.863	0.553	0.979	0.854	0.896	0.931	0.827
	Gemini-2.5 Flash	0.890	0.912	0.269	0.912	0.922	0.996	0.971	0.881	0.804	0.619	0.769	0.648	0.961	0.929	0.854	0.936	0.830
MZS+B	GPT-4.1	0.840	0.908	0.377	0.944	0.931	0.994	0.975	0.880	0.822	0.597	0.853	0.624	0.990	0.899	0.897	0.930	0.841
	GPT-4o-mini	0.879	0.920	0.287	0.957	0.906	0.993	0.969	0.926	0.833	0.637	0.902	0.738	0.961	0.890	0.955	0.943	0.856
	Gemini-2.5 Flash	0.871	0.947	0.338	0.914	0.922	0.996	0.975	0.866	0.808	0.637	0.785	0.549	0.986	0.910	0.854	0.926	0.830
Ours	GPT-4.1	0.935	0.925	0.918	0.989	0.985	0.987	0.983	0.788	0.774	0.603	0.908	0.784	0.884	0.963	0.836	0.849	0.882
	GPT-4o-mini	0.942	0.930	0.903	0.990	0.971	0.983	0.983	0.815	0.814	0.618	0.940	0.814	0.905	0.970	0.884	0.854	0.895
	Gemini-2.5 Flash	0.975	0.950	0.897	0.994	0.989	0.993	0.988	0.860	0.841	0.605	0.885	0.779	0.929	0.981	0.888	0.892	0.903

Table 1: Criterion-level results across pipelines, generation models, and task types, averaged across the three evaluator models. SAQ criteria: OG-T (objective gradability, triple-aware), OG-S (objective gradability, stem-only), MSQ (mark scheme quality), AF (answer fidelity), STI (single-task integrity), Cla. (clarity), and CG (course grounding). SBQ extends SAQ with two scenario-specific criteria: SRN (scenario relevance/necessity) and SG (scenario grounding). Avg is the macro-average over all 16 non-Bloom criteria. **Bold** = best per column; underline = second.

Pipe.	SAQs									SBQs									Avg
	GPT-4.1			GPT-4o-mini			Gemini-2.5 Flash			GPT-4.1			GPT-4o-mini			Gemini-2.5 Flash			
	BLA	BA	HOSR	BLA	BA	HOSR	BLA	BA	HOSR	BLA	BA	HOSR	BLA	BA	HOSR	BLA	BA	HOSR	
ZS	0.781	0.019	0.000	0.747	0.057	0.000	0.771	0.000	0.000	0.775	0.091	0.111	0.759	0.205	0.333	0.784	0.270	0.000	0.317
ZS+B	0.894	0.736	0.750	0.891	0.679	0.500	0.889	0.694	0.357	0.864	0.576	0.640	0.894	0.676	0.833	0.894	0.694	0.636	0.728
MZS	0.765	0.042	0.045	0.795	0.075	0.158	0.772	0.071	0.000	0.745	0.089	0.074	0.767	0.181	0.167	0.745	0.137	0.000	0.313
MZS+B	0.885	0.637	0.750	0.899	0.681	0.727	0.899	0.722	0.750	0.853	0.558	0.641	0.879	0.625	0.815	0.850	0.514	0.467	0.731
Ours	0.985	0.947	0.919	0.997	0.991	0.989	0.961	0.880	0.884	0.981	0.938	0.943	0.988	0.964	0.975	0.939	0.847	0.811	0.941

Table 2: Bloom-related metrics across pipelines, generation models, and assessment formats. Columns are grouped by assessment format and generation model, and rows correspond to pipelines. BLA denotes the soft Bloom alignment score, while BA and HOSR are derived from the aggregated inferred Bloom level for each item. BA denotes exact Bloom alignment accuracy, and HOSR denotes higher-order success rate. Avg is the macro-average over all displayed Bloom metrics across formats and generators. Higher is better for all metrics. **Bold** = best per column; underline = second best.

across all pipeline-generator configurations and both assessment formats. Overall, the proposed pipeline remains the strongest system across the main evaluation settings. Its advantage is clearest for SAQs, where it dominates both criterion-level quality and pairwise win rates. For SBQs, the proposed pipeline is strongest in Bloom control and scenario grounding, but it does not uniformly dominate simpler baselines on all other criteria. In particular, objective gradability and answer fidelity remain competitive across pipelines, suggesting that simpler scenario-based items can sometimes be easier to grade and align more directly with their expected answers.

Assessment Quality The clearest criterion-level separation is mark scheme quality (Table 1). For SAQs, the proposed pipeline achieves MSQ scores of 0.897–0.918 across generators, compared with 0.236–0.519 for the baselines, indicating that the decomposition and verification stages substantially

improve the coverage and structure of marking points. The same pattern holds for SBQs, although with a smaller margin: the proposed pipeline reaches 0.779–0.814, while the strongest baseline reaches 0.738.

Other criteria show a more nuanced pattern. For SAQs, the proposed pipeline remains consistently strong on AF (0.989–0.994), STI (0.971–0.989), and CG (0.983–0.988), suggesting that the generated answers, mark schemes, and source grounding remain internally consistent. However, for SBQs, the proposed pipeline does not uniformly dominate simpler baselines on all non-Bloom criteria. Some baselines obtain higher OG-T, OG-S, and AF scores, suggesting that the added contextual complexity of scenarios can make objective gradability and answer alignment harder to satisfy consistently. Overall, the proposed pipeline improves the more structured dimensions of assessment construction, especially MSQ and scenario grounding,

Fmt.	Pipe.	<i>n</i>	Rub. Ag.	BLA Ag.	Std.	<i>r</i>	Lvl Ex./Pair
SAQ	ZS	140	0.773	0.864	0.060	0.314	0.636 / 0.750
	ZS+B	142	0.774	0.725	0.059	0.394	0.718 / 0.807
	MZS	261	0.724	0.782	0.072	0.436	0.663 / 0.768
	MZS+B	228	0.747	0.693	0.068	0.384	0.684 / 0.782
	Ours	1072	0.876	0.880	0.033	0.356	0.878 / 0.916
SBQ	ZS	136	0.645	0.640	0.098	0.445	0.529 / 0.669
	ZS+B	129	0.652	0.597	0.101	0.472	0.597 / 0.716
	MZS	246	0.615	0.622	0.109	0.382	0.524 / 0.667
	MZS+B	214	0.633	0.537	0.107	0.388	0.500 / 0.653
	Ours	750	0.667	0.860	0.098	0.484	0.860 / 0.905

Table 3: Inter-judge agreement across GPT-4o-mini, GPT-4.1-mini, and Gemini-2.5-Flash by assessment format and pipeline. Rub. Ag. is macro-average exact agreement over non-Bloom rubric criteria. BLA Ag. is exact agreement over the discretized soft Bloom-alignment score. Std. is the macro-average standard deviation of normalized non-Bloom rubric scores; *r* is mean pairwise Pearson correlation over non-Bloom rubric scores. Lvl Ex./Pair reports exact three-judge agreement / mean pairwise agreement on the inferred Bloom-level label. **Bold** = best per column; underline = second best.

while SBQ generation remains intrinsically more challenging.

Bloom Cognitive Control The strongest and most consistent gains occur in Bloom-level control (Table 2). Pipelines without explicit Bloom conditioning have very low exact Bloom accuracy across generators: ZS reaches BA values of only 0.000–0.057 for SAQs and 0.091–0.270 for SBQs, while MZS remains similarly low at 0.042–0.075 and 0.089–0.181 respectively. This indicates that decomposition alone does not provide reliable cognitive-level control.

Adding Bloom prompts substantially improves alignment, but remains less reliable than the proposed pipeline. ZS+B and MZS+B improve BA across both formats, but the proposed pipeline achieves the highest Bloom metrics for every generator and assessment format. For SAQs, it reaches BA values of 0.880–0.991 and HOSR values of 0.884–0.989; for SBQs, it reaches BA values of 0.847–0.964 and HOSR values of 0.811–0.975.

The macro-average column further shows that this improvement is not driven by a single generator or format. The proposed pipeline achieves the highest overall Bloom average, 0.941, compared with 0.731 for MZS+B and 0.728 for ZS+B. This suggests that Bloom-specialized generation and verification improve cognitive targeting across model families, especially for higher-order levels where prompt-only conditioning is less consistent.

Scenario Construction For SBQs, scenario grounding is highest under the proposed pipeline across all generators, with SG scores of 0.908,

Fmt.	Crit.	Agree	Std.	<i>r</i>
SAQ	OG-T	0.662	0.086	0.250
	OG-S	0.649	0.083	0.266
	MSQ	0.753	0.081	0.740
	AF	0.868	0.043	0.314
	STI	0.635	0.086	0.290
	Cla.	0.954	0.011	0.189
	CG	0.931	0.017	0.587
SBQ	BLA	0.789	0.032	<u>0.595</u>
	OG-T	0.602	0.129	0.388
	OG-S	0.500	0.144	0.365
	SRN	<u>0.321</u>	0.166	0.302
	SG	0.650	0.085	0.455
	MSQ	0.711	0.099	0.723
	AF	0.925	0.029	0.310
	STI	0.580	0.101	0.348
	Cla.	0.670	0.124	0.421
	CG	<u>0.823</u>	0.050	<u>0.595</u>
	BLA	0.651	<u>0.047</u>	0.570

Table 4: Criterion-level inter-judge agreement, macro-averaged over pipelines. Agree is exact three-judge agreement; Std. is the mean standard deviation of normalized scores across judges; *r* is the mean pairwise Pearson correlation. For non-Bloom criteria, values are computed over rubric scores. For BLA, values are computed separately over the discretized soft Bloom-alignment score. SAQ criteria: OG-T, OG-S, MSQ, AF, STI, Cla., CG, and BLA. SBQ additionally includes SRN and SG. **Bold** = best per column within each format; underline = second best.

0.940, and 0.885 for GPT-4.1, GPT-4o-mini, and Gemini-2.5 Flash respectively (Table 1). This supports the value of the dedicated scenario generation, critique, and refinement stages in anchoring scenarios to the source material.

However, SRN remains less clearly improved. The proposed pipeline scores 0.603–0.618, while baselines range from 0.551 to 0.681, suggesting that the pipeline improves scenario grounding more reliably than scenario necessity (Table 1). The agreement results support this interpretation: SRN has the lowest SBQ judge agreement (0.321) and highest standard deviation (0.166), indicating that scenario necessity is both difficult to generate and difficult to judge consistently (Table 4).

Inter-Judge Agreement At the pipeline level, the proposed method achieves the highest rubric agreement, discretized BLA agreement, and inferred Bloom-level agreement for both formats (Table 3). For SAQs, these are 0.876, 0.880, and 0.878/0.916 exact/pairwise agreement respectively; for SBQs, they are 0.667, 0.860, and 0.860/0.905. This strengthens the Bloom-control claim, as the proposed pipeline achieves both the highest Bloom scores and the most consistent evaluator agreement on inferred cognitive levels.

Agreement is generally lower for SBQs than SAQs, reflecting the added subjectivity of scenario relevance, grounding, and context-dependent grading. Criterion-level results further clarify where

Pipeline	Generator	SAQ Win Rate		SBQ Win Rate	
		Raw	Production	Raw	Production
ZS	GPT-4o	0.42 [0.30, 0.54]	0.47 [0.35, 0.58]	0.40 [0.20, 0.62]	0.17 [0.07, 0.28]
	Gemini	<u>0.48</u> [0.29, 0.67]	0.27 [0.10, 0.46]	0.46 [0.34, 0.59]	0.34 [0.24, 0.44]
	GPT-4.1	0.39 [0.25, 0.51]	0.30 [0.20, 0.41]	0.41 [0.31, 0.50]	0.32 [0.23, 0.43]
ZS+B	GPT-4o	0.41 [0.30, 0.52]	0.28 [0.19, 0.38]	0.50 [0.38, 0.62]	0.42 [0.23, 0.62]
	Gemini	0.46 [0.35, 0.56]	0.38 [0.23, 0.52]	0.60 [0.45, 0.75]	0.40 [0.25, 0.55]
	GPT-4.1	0.42 [0.30, 0.55]	<u>0.47</u> [0.35, 0.58]	<u>0.60</u> [0.47, 0.75]	0.41 [0.31, 0.53]
MZS	GPT-4o	0.23 [0.14, 0.31]	0.31 [0.21, 0.42]	0.42 [0.20, 0.62]	0.38 [0.20, 0.57]
	Gemini	0.29 [0.12, 0.50]	0.27 [0.15, 0.42]	0.47 [0.30, 0.65]	0.40 [0.28, 0.54]
	GPT-4.1	0.28 [0.19, 0.39]	0.27 [0.18, 0.38]	0.30 [0.19, 0.42]	0.32 [0.21, 0.45]
MZS+B	GPT-4o	<u>0.44</u> [0.32, 0.57]	0.46 [0.35, 0.57]	<u>0.55</u> [0.28, 0.80]	<u>0.68</u> [0.47, 0.88]
	Gemini	0.46 [0.27, 0.65]	<u>0.58</u> [0.48, 0.67]	0.35 [0.20, 0.50]	<u>0.47</u> [0.31, 0.65]
	GPT-4.1	<u>0.47</u> [0.35, 0.57]	<u>0.47</u> [0.35, 0.57]	0.44 [0.31, 0.56]	<u>0.52</u> [0.41, 0.63]
Ours	GPT-4o	0.95 [0.89, 0.99]	0.97 [0.92, 1.00]	0.62 [0.40, 0.85]	0.78 [0.65, 0.88]
	Gemini	0.79 [0.56, 0.98]	0.94 [0.88, 1.00]	0.60 [0.44, 0.76]	0.86 [0.78, 0.94]
	GPT-4.1	0.92 [0.83, 0.99]	0.94 [0.89, 0.99]	0.75 [0.62, 0.86]	0.92 [0.85, 0.97]

Table 5: Per-generator matched-task win rates with 95% bootstrap confidence intervals for SAQs and SBQs under raw and production selection modes. Win rates are computed against the pool of all other pipelines using discrimination-weighted scores aggregated across the three evaluator models. Raw mode uses the first generated candidate; production mode selects the highest-scoring candidate. GPT-4o denotes GPT-4o-mini; Gemini denotes Gemini-2.5-Flash. **Bold** = best per generator and evaluation mode; underline = second best.

agreement is most stable (Table 4). For SAQs, judges agree most strongly on clarity (0.954), course grounding (0.931), and answer fidelity (0.868), while MSQ has the highest pairwise correlation ($r = 0.740$). For SBQs, answer fidelity has the highest agreement (0.925), and MSQ again has the highest pairwise correlation ($r = 0.723$). By contrast, SRN has the lowest SBQ agreement (0.321), confirming that scenario necessity is the least stable judgement dimension.

Pairwise Outcomes Matched-task pairwise comparisons support the same overall trend (Table 5). For SAQs, the proposed pipeline achieves the highest win rates under every generator and selection mode. In raw mode, it obtains win rates of 0.95, 0.79, and 0.92 for GPT-4o-mini, Gemini-2.5 Flash, and GPT-4.1 respectively. In production mode, these increase to 0.97, 0.94, and 0.94. This indicates that the proposed pipeline improves both intrinsic generation quality and the quality of the criterion-based selected output.

For SBQs, the proposed pipeline achieves the strongest production-mode results, with win rates of 0.78, 0.86, and 0.92 across the three generators. Raw SBQ performance is less decisive, with win rates of 0.62, 0.60, and 0.75, reflecting the variability introduced by SBQs. The stronger production-mode results suggest that future deployment could benefit from criterion-based candidate selection, particularly for SBQs, where scenario-conditioned

generation introduces greater variability.

Finally, the proposed pipeline exhibits a yield advantage in the retained item pool, with 1072 SAQs and 750 SBQs compared with the largest baseline pools of 261 and 246, respectively. Together with its stronger Bloom-control results, this suggests that the framework produces more valid assessment tuples while producing stronger alignment with the intended cognitive levels.

5 Conclusion

We presented a multi-agent, multi-stage framework for generating structured assessment items with Bloom-level cognitive control and automated verification. Across five pipelines, the results show that prompt-level Bloom conditioning improves alignment but remains less reliable than Bloom-specialized few-shot generation with staged verification. The proposed pipeline achieves the strongest overall performance, particularly in Bloom alignment, higher-order cognitive targeting, mark scheme quality, and valid-item yield. The results also show that SBQs remain more challenging than SAQs: scenario grounding improves, but scenario necessity, objective gradability, and answer fidelity are harder to satisfy consistently. Overall, reliable assessment generation appears to require pipeline-level specialization and validation rather than generic zero-shot or prompt-level Bloom conditioning alone.

Limitations

Our evaluation is conducted on a single domain: university-level Artificial Intelligence course material. Although the framework is intended to be domain-agnostic, we have not verified whether the same results generalize to disciplines with substantially different assessment conventions, such as medicine, law, humanities, or mathematically intensive subjects. Mathematical, numerical, proof-based, and programming-oriented questions may also require stricter verification than the rubric-based checks used here, since correct answers may involve deterministic values, multiple valid solution paths, or executable code. Therefore, future work should evaluate these item types separately, potentially combining rubric-guided judging with symbolic checking, unit tests, code execution, or answer-equivalence verification.

Although we complement automated evaluation with structured blind human assessment, the manual study covers only 18 matched-task comparisons across five pipelines. This provides useful qualitative validation, but limits the statistical conclusions that can be drawn. A larger-scale human evaluation with multiple annotators and inter-annotator agreement measures would more robustly validate the pedagogical quality of the generated items. Our automated evaluation uses three LLM-as-a-judge models rather than a single judge, but remains an imperfect proxy for pedagogical quality. The agreement analysis shows that Bloom-related measures and mark scheme quality are relatively more stable, whereas scenario relevance and necessity (SRN) are less stable. Thus, conclusions about Bloom control and mark scheme quality are better supported than conclusions about whether a scenario is genuinely necessary for answering an SBQ. Future work should refine the SRN rubric and validate this criterion more extensively with human assessors.

We evaluate proprietary API-based models only, using GPT-4.1, GPT-4o-mini, and Gemini-2.5 Flash as generators, and GPT-4o-mini, GPT-4.1-mini, and Gemini-2.5 Flash as evaluators. These models were chosen as a practical cost-efficiency trade-off, since the evaluation required scoring many generated items across multiple criteria. However, the framework is not tied to proprietary models. Future work should evaluate open-source generators and judges, including smaller models, to study model-size scaling, cost-performance trade-offs, and deployment in resource-constrained educational settings. Stronger evaluator models may also improve verification for reasoning-heavy or

domain-specialist assessment items.

The generation and evaluation models are not fully independent, since GPT-family models appear in both roles, and Gemini is used as both a generator and evaluator. We reduce this risk by reporting results across multiple generator-evaluator configurations and including models from more than one provider, but a broader judge ensemble would further reduce the risk of model-family bias. We also do not explore heterogeneous pipeline configurations, such as using different models for question generation, scenario generation, mark scheme generation, answer generation, and verification.

The proposed pipeline is evaluated as an integrated framework, rather than as a controlled test of any individual component. It differs from the baselines through a combination of Bloom-specialized few-shot prompting, level-specific mark scheme templates, staged generation, scenario critique and refinement for SBQs, and generation-time verification. These choices are intended to work together, but the present experiments do not isolate their individual contributions. Future work should conduct component-level ablations, such as removing in-context examples, disabling verification, replacing Bloom-specialized prompts with generic Bloom prompts, or removing Bloom-aware mark scheme templates. Full specifications of each pipeline component are provided in Appendix A.

Finally, the yield advantage of the proposed pipeline comes from generating candidates for multiple Bloom levels and filtering them through verification, which increases computational cost. The framework does not use open-ended agent interaction. Each item passes through a fixed sequence of generation: critique, refinement, and verification stages, with failed candidates being discarded rather than recursively regenerated until they pass. The SAQ and SBQ acceptance predicates, including Boolean decision variables, JSON decision fields, and check logic, are specified in Appendix A.6. This design prevents unbounded verifier-generator loops and keeps execution predictable, but lowers the potential yield from the provided source material. We also do not analyse latency, cost, verifier failure rates, or alternative retry strategies in detail. Future work should study these trade-offs more systematically. The validated assessment tuples produced by the framework could also provide supervision for fine-tuning models that generate Bloom-aligned items directly from source context, potentially reducing the need for expensive multi-stage generation at inference.

References

- Kuang Wen Chan, Farhan Ali, Joonhyeong Park, Kah Shen Brandon Sham, Erdalyn Yeh Thong Tan, Francis Woon Chien Chong, Kun Qian, and Guan Kheng Sze. 2025. [Automatic item generation in various STEM subjects using large language model \(LLM\) prompting](#). *Computers and Education: Artificial Intelligence*, 8:100344.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. [Automatic question generation and answer assessment: a survey](#). *Research and Practice in Technology Enhanced Learning*, 16(1):5.
- Weiping Fu, Bifan Wei, Jianxiang Hu, Zhongmin Cai, and Jun Liu. 2024. [Qgeval: Benchmarking multi-dimensional evaluation for question generation](#). *Preprint*, arXiv:2406.05707.
- Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. [Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 13806–13834. Association for Computational Linguistics.
- Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Solafa Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:1–70.
- Jiazheng Li, Artem Bobrov, David West, Cesare Aloisi, and Yulan He. 2024a. [An automated explainable educational assessment system built on llms](#). *Preprint*, arXiv:2412.13381.
- Jiazheng Li, Lin Gui, Yuxiang Zhou, David West, Cesare Aloisi, and Yulan He. 2023. [Distilling ChatGPT for explainable automated student answer assessment](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6007–6026, Singapore. Association for Computational Linguistics.
- Jiazheng Li, Hainiu Xu, Zhaoyue Sun, Yuxiang Zhou, David West, Cesare Aloisi, and Yulan He. 2024b. [Calibrating LLMs with preference optimization on thought trees for generating rationale in science question scoring](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5452–5479, Miami, Florida, USA. Association for Computational Linguistics.
- Jiazheng Li, Yuxiang Zhou, Junru Lu, Gladys Tyen, Lin Gui, Cesare Aloisi, and Yulan He. 2025. [Two heads are better than one: Dual-model verbal reflection at inference-time](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3119–3140, Suzhou, China. 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](#). *Preprint*, arXiv:2303.16634.
- Anderson LW, Krathwohl DR, Airasian PW, Cruikshank KA, Richard Mayer, Pintrich PR, J. Rath, and Witrock MC. 2001. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). *Preprint*, arXiv:2303.17651.
- Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. 2024. [How effective is gpt-4 turbo in generating school-level questions from textbooks based on bloom’s revised taxonomy?](#) *Preprint*, arXiv:2406.15211.
- Alireza Mohammadshahi, Thomas Scialom, Majid Yazdani, Pouya Yanki, Angela Fan, James Henderson, and Marzieh Saeidi. 2023. [Rqge: Reference-free metric for evaluating question generation by answering the question](#). *Preprint*, arXiv:2211.01482.
- Bang Nguyen, Mengxia Yu, Yun Huang, and Meng Jiang. 2024. [Reference-based metrics disprove themselves in question generation](#). *Preprint*, arXiv:2403.12242.
- Shinhyeok Oh, Hyojun Go, Hyeongdon Moon, Yunsung Lee, Myeongho Jeong, Hyun Seung Lee, and Seungtaek Choi. 2023. [Evaluation of question generation needs more references](#). *Preprint*, arXiv:2305.16626.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024. [Automated educational question generation at different bloom’s skill levels using large language models: Strategies and evaluation](#). In *Proceedings of the 25th International Conference on Artificial Intelligence in Education (AIED)*. ArXiv preprint arXiv:2408.04394.

Y. Song, J. Du, and Q. Zheng. 2025. [Automatic item generation for educational assessments: a systematic literature review](#). *Interactive Learning Environments*, 33(9):5386–5405.

Bin Tan, Nour Armoush, Elisabetta Mazzullo, Okan Bulut, and Mark Gierl. 2025. [A review of automatic item generation techniques leveraging large language models \(LLMs\)](#). *International Journal of Assessment Tools in Education*, 12(2):317–340.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. [Gemini: A family of highly capable multimodal models](#). *Preprint*, arXiv:2312.11805.

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.

Antoun Yaacoub, Jérôme Da-Rugna, and Zainab As-saghir. 2025. [Assessing ai-generated questions’ alignment with cognitive frameworks in educational assessment](#). *International Journal of Computer Theory and Engineering*, 17(3):114–125.

Archana Yadav, Harshvivek Kashid, Medchalimi Sruthi, B JayaPrakash, Chintalapalli Raja Kullayappa, Mandala Jagadeesh Reddy, and Pushpak Bhattacharyya. 2025. [From recall to creation: Generating follow-up questions using bloom’s taxonomy and grice’s maxims](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1322–1338, Vienna, Austria. Association for Computational Linguistics.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Trans. Inf. Syst.*

Runcong Zhao, Artem Bobrov, Jiazheng Li, Cesare Aloisi, and Yulan He. 2025. [LearnLens: LLM-enabled personalised, curriculum-grounded feedback with educators in the loop](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 625–633, Suzhou, China. Association for Computational Linguistics.

Code repository. Code and supporting materials for the generation and evaluation pipeline are available at: <https://github.com/LyleW473/BloomAssessmentItemGeneration>.

A Generation Pipeline Details

A.1 Pipelines & Prompt Templates

This section describes the generation-time pipeline used to produce assessment items. All pipelines produce structured JSON outputs and return generated items as either `valid_questions` or `failed_questions`. Across the proposed pipeline, difficulty is not independently assigned by the model; instead, it is deterministically derived from the target Bloom level: knowledge and understanding map to easy, application and analyze map to medium, and synthesis and evaluation map to hard.

A.2 Baseline Generation Pipelines

The zero-shot baseline (ZS) generates the full assessment tuple in a single LLM call. For SAQs, the model outputs a question, difficulty, mark scheme, total marks, and expected answer. For SBQs, the same call additionally generates a scenario. The zero-shot Bloom baseline (ZS+B) uses the same single-stage structure, but augments the system prompt with a Bloom taxonomy definition block and asks the model to assign a `bloom_level` to each item.

The multi-stage zero-shot baseline (MZS) decomposes generation into three sequential calls: question generation, mark scheme generation, and expected answer generation. For SBQs, the first stage generates both the scenario and the question. The multi-stage Bloom baseline (MZS+B) adds Bloom taxonomy definitions to the first-stage question generation prompt and records the model-assigned `bloom_level`; however, this Bloom label does not directly control the later mark scheme or answer generation stages.

A.3 Proposed Generation Pipeline

The proposed pipeline differs from the baselines in three main ways. First, generation is driven by the target Bloom level rather than relying on the model to self-assign a level. The pipeline iterates over all six Bloom levels and invokes a level-specific question-generation prompt for each level. Second, these Bloom-level prompts are few-shot templates, each containing level-specific instructions, allowed and prohibited question forms, output-format constraints, and worked examples illustrating the intended cognitive demand. In contrast, the baseline prompts specify task instructions and output schemas, but do not include content-filled in-context demonstrations. Third, generation

is interleaved with verification, where candidates that fail Bloom alignment, grounding, mark scheme coverage, or answer coverage checks are discarded rather than passed to later stages.

For SAQs, the proposed pipeline follows the structure:

- For each Bloom level $b \in \mathcal{B}$:
 - Generate question stems from the source content using the prompt for b .
 - Verify Bloom alignment and course relevance.
 - Generate a Bloom-aware mark scheme.
 - Verify mark scheme coverage.
 - Generate the expected answer.
 - Verify answer coverage and keep valid items.

For SBQs, the proposed pipeline follows the structure:

- Generate an initial scenario from the source content.
- Critique the scenario for compliance.
- Refine the scenario if required.
- For each Bloom level $b \in \mathcal{B}$:
 - Generate SBQ stems from the refined scenario using the prompt for b .
 - Verify Bloom alignment and scenario relevance.
 - Generate a Bloom-aware mark scheme.
 - Verify mark scheme coverage with respect to the scenario.
 - Generate the expected answer.
 - Verify answer coverage and keep valid items.

The SBQ pipeline therefore extends the SAQ tuple-generation process with a scenario generation–critique–refinement stage. After refinement, the scenario is used as the factual input for Bloom-level question generation, mark scheme generation, answer generation, and verification. This means that SBQs use the same Bloom-controlled generation logic as SAQs, but all downstream stages are conditioned on the generated scenario rather than directly on the source text.

A.4 Bloom-Level Question Generation Prompts

In the proposed pipeline, question generation is controlled by separate Bloom-level system prompts rather than a single generic Bloom instruction. Each Bloom level has its own prompt, but all prompts follow the same template structure. The

Bloom Level	Cognitive Goal	Representative Verbs
Knowledge	Recall facts and terms	define, identify, list, state
Understanding	Explain or interpret meaning	explain, summarize, interpret, illustrate
Application	Use knowledge in a new case	apply, compute, solve, demonstrate
Analyze	Break down relationships	analyze, compare, differentiate, infer
Evaluation	Make justified judgements	assess, defend, evaluate, justify
Synthesis	Construct a new whole	design, formulate, generate, plan

Table 6: Summary of the Bloom-level prompt specialization used in the proposed pipeline. Each level uses a dedicated prompt with level-specific goals, allowed verbs, prohibited stems, and examples.

prompt first assigns the model the role of an expert educational content generator specializing in a specific Bloom level, then defines the cognitive operation that questions at that level should require. It then specifies the input format, output JSON schema, allowed command verbs, prohibited lower- or higher-level stems, example question forms, and strict output formatting requirements.

The generic structure of each Bloom-level prompt is:

- **Role:** Assigns the model the role of an expert generator for a specific Bloom level.
- **Goal:** Defines the target cognitive operation required by that Bloom level.
- **Input:** Specifies the source content and target Bloom level.
- **Output:** Requires a JSON array of question objects.
- **Guidelines:** Lists allowed verbs and the expected reasoning depth.
- **Restrictions:** Prohibits off-level stems and lower- or higher-level phrasing.
- **Examples:** Provides level-specific examples of valid question forms.
- **Formatting:** Requires valid JSON only, with no additional commentary.

Thus, the same Bloom-specialized prompt structure is used for both formats, but SAQs generate questions directly from course content, while SBQs generate questions from the refined scenario.

The SBQ application-level prompt includes an additional eligibility constraint. An application-level SBQ is generated only if the scenario contains sufficient numerical values, counts, or structured outcomes to support the application of methods. If this information is absent, the model is instructed to return an empty array. This prevents computationally framed questions from being generated when the scenario does not contain the information necessary to solve them.

Bloom Level	Points	Marks/Point	Total Marks
Knowledge	1–3	1	1–3
Understanding	2–4	1–2	3–5
Application	3–5	1–2	4–8
Analyze	4–6	1–2	6–10
Evaluation	4–6	1–2	6–10
Synthesis	5–8	1–3	8–12

Table 7: Bloom-level mark scheme templates used by the proposed pipeline. The model must output a single integer mark value for each point; range strings such as "1–2" are not allowed.

A.5 Bloom-Aware Mark Scheme Generation

The proposed pipeline also uses a Bloom-aware few-shot mark scheme prompt. Unlike the baseline mark scheme prompts, which provide only general marking guidance and schema constraints, this prompt conditions the structure of the mark scheme on the target Bloom level and includes worked examples showing how marking points should differ across cognitive levels. The input specifies the question, difficulty, Bloom level, and source content, and the model is instructed to return a structured JSON mark scheme with one integer mark value per point.

The prompt constrains both the *structure* and *content* of the mark scheme. Structurally, each Bloom level is assigned an expected range of marking points, marks per point, and total marks. Content-wise, lower Bloom levels require concise factual marking points, while higher Bloom levels require points covering reasoning, comparison, judgement, design components, or justification. Thus, the mark scheme is not generated as a generic list of possible answer points, but is explicitly shaped by the intended cognitive demand of the question through both level-specific constraints and few-shot exemplars.

The prompt further specifies level-specific content expectations:

- **Knowledge:** Strictly factual points with no elaboration.
- **Understanding:** Light explanatory points without deep reasoning.
- **Application:** Concrete procedural or method-application steps.
- **Analyze:** Structural, relational, or comparative points.
- **Evaluation:** Criteria-based judgement points.
- **Synthesis:** Design components, justification, and rationale.

Difficulty affects the expected depth of each marking point, but not the Bloom-level point-count or mark-allocation template. Easy items require minimal depth, medium items require moderate interconnected detail, and hard items require more precise technical explanation. After generation, the implementation recomputes `total_marks` as the sum of the generated point values, rather than relying on the model-provided total. This enforces consistency between the mark allocation and the reported total score, and reduces the number of valid candidates being rejected due to simple arithmetic errors in the generated total.

A.6 Generation-Time Verification Predicate

The proposed pipeline uses a verification predicate during generation to decide whether a candidate assessment tuple is retained. This generation-time filtering is distinct from the final evaluation protocol in Section B. Verification determines whether a candidate is passed into the generated dataset, whereas evaluation scores the items within the generated dataset.

All verification checks are implemented as binary rubric-guided LLM calls that return structured JSON.

Design principle. There are no numeric generation-time thresholds. Each verifier returns a strict Boolean decision via a named JSON field (see Table 8), and a single False at any stage rejects the candidate tuple, writing it to the failed-output pool. All checks are short-circuiting: once a candidate fails, no subsequent checks are run for that item.

For SAQs, a candidate tuple is accepted only if it passes all four checks:

$$\text{ACCEPT}(q, m, a) \iff v_{\text{Bloom}} \wedge v_{\text{rel}} \wedge v_{\text{ms}} \wedge v_{\text{ans}},$$

where q is the question, m the mark scheme, and a the expected answer. The four decision variables correspond to Bloom-level alignment, course relevance, mark scheme coverage, and expected answer coverage.

For SBQs, the same tuple-level predicate is applied after an additional scenario-compliance stage. A scenario s is first generated, critiqued for compliance, and conditionally refined to produce a final scenario s' . The candidate SBQ tuple is then accepted only if the question, mark scheme, and expected answer pass the same four checks with respect to s' :

$$\text{ACCEPT}_{\text{SBQ}}(s', q, m, a) \iff v_{\text{Bloom}} \wedge v_{\text{rel}}^s \wedge v_{\text{ms}}^s \wedge v_{\text{ans}}.$$

Check	Input	Decision field	Failure action
Bloom alignment	Question, target Bloom level	belongs_to_level	Reject
Course/scenario relevance	Question, source content or scenario	is_relevant	Reject
Mark scheme coverage	Question, mark scheme, Bloom level, content	is_correct	Reject
Expected answer coverage	Question, mark scheme, expected answer	is_covered	Reject

Table 8: Summary of generation-time verification checks. All checks return a Boolean decision and a short reason.

Bloom level	Difficulty
Knowledge	easy
Understanding	easy
Application	medium
Analyze	medium
Evaluation	hard
Synthesis	hard

Table 9: Deterministic mapping from target Bloom level to item difficulty.

Here, v_{rel}^s verifies that the question is grounded in the refined scenario, and v_{ms}^s verifies that the mark scheme is also grounded in that scenario. Thus, SBQs use the same acceptance logic as SAQs, but with an additional scenario generation–critique–refinement stage and scenario-grounded relevance checks.

Difficulty is derived deterministically from the target Bloom level rather than assigned independently. We use the mapping in Table 9 for mark scheme generation and subsequent verification.

SAQ verification. For each source document, the SAQ pipeline iterates over all Bloom levels. It first generates question stems using Bloom-specialized system instructions. Each stem is then verified for Bloom alignment and course relevance. Only stems passing both phase-one checks proceed to mark scheme generation. The generated mark scheme is then checked for coverage, factual correctness, Bloom-level structural compliance, and alignment with the question. If this check passes, an expected answer is generated and verified against the mark scheme. Only candidates passing all four checks are added to the valid-output pool.

The Bloom alignment check asks the verifier to infer the true cognitive level of the question from its phrasing and cognitive demand, then compare it with the declared target level. The course relevance check verifies that the question can be answered from substantive course content rather than from administrative or other external information. The mark scheme coverage check verifies that the marking points are factually accurate, complete, aligned with the question, and consistent with the required Bloom-level structure. The expected answer coverage check verifies that the prose answer expresses every mark scheme point at the required cognitive

Bloom level	Points	Marks/point	Total marks
Knowledge	1–3	1	1–3
Understanding	2–4	1–2	3–5
Application	3–5	1–2	4–8
Analyze	4–6	1–2	6–10
Evaluation	4–6	1–2	6–10
Synthesis	5–8	1–3	8–12

Table 10: Bloom-level structural templates used for mark scheme generation and verification.

depth without contradiction.

SBQ verification. The SBQ pipeline adds a scenario-compliance phase before the four question-level checks. First, a neutral factual scenario is generated from the source content. A critique agent then reviews the scenario for seven violation types: metric-as-conclusion, teaching or definition, interpretation or judgement, mental-state or goal language, rationale or intent, recommendation or next step, and question leakage. If the scenario is compliant, it is used directly. If not, a refinement step applies the minimal changes required to remove the detected violations.

After scenario refinement, SBQ question generation and verification proceed as in the SAQ pipeline. The key difference is that relevance and mark scheme grounding are checked against the refined scenario rather than the full source document. This ensures that the question and mark scheme are grounded in the scenario actually presented to the student.

The framework therefore uses a fixed-stage verification process rather than open-ended agent interaction. Failed question or tuple candidates are discarded rather than recursively regenerated until they pass, which keeps execution predictable and prevents unbounded verifier–generator loops. However, this design also means that the system may sacrifice potential yield from source material when otherwise recoverable candidates fail a verification stage.

B Evaluation Details

B.1 Bloom-Specific Metrics

Let $\hat{b}_i$ denote the inferred Bloom level of generated item i , and b_i its corresponding target level. For a given pipeline p and generation model g , Bloom alignment accuracy is defined as

$$\text{BA}^{(p,g)} = \frac{1}{|\mathcal{D}^{(p,g)}|} \sum_{i \in \mathcal{D}^{(p,g)}} I(\hat{b}_i = b_i),$$

which measures the proportion of generated questions whose inferred cognitive level matches the intended one.

To isolate the challenge of generating higher-order questions, we restrict evaluation to $\mathcal{B}_{\text{high}} = \{\text{analysis, evaluation, synthesis}\}$ and define the higher-order success rate as

$$\text{HOSR}^{(p,g)} = \frac{1}{|\mathcal{D}_{\text{high}}^{(p,g)}|} \sum_{i \in \mathcal{D}_{\text{high}}^{(p,g)}} I(\hat{b}_i = b_i),$$

where $\mathcal{D}_{\text{high}}^{(p,g)}$ denotes the subset of questions whose target Bloom level lies in $\mathcal{B}_{\text{high}}$.

B.2 Criterion-Based Scoring

Given a pipeline $p \in \mathcal{P}$, a generation model $g \in \mathcal{G}$, and a set of evaluation criteria $\mathcal{C}$, each generated item i is assigned a discrete score $s_{i,c}^{(p,g)} \in \{0, 1, 2\}$ for criterion $c \in \mathcal{C}$ using a rubric-guided LLM judge. Scores are normalized to the unit interval as $\tilde{s}_{i,c}^{(p,g)} = s_{i,c}^{(p,g)} / 2$.

Aggregate performance is characterized by the mean normalized score per criterion:

$$\mu_c^{(p,g)} = \frac{1}{N_{p,g}} \sum_{i=1}^{N_{p,g}} \tilde{s}_{i,c}^{(p,g)}.$$

B.3 Discrimination-Weighted Aggregation

To obtain a single scalar score per item, we introduce a discrimination-weighted aggregation scheme that emphasizes criteria which better separate pipeline performance.

For each criterion $c \in \mathcal{C}$, we first compute the mean score per pipeline by averaging across generation models:

$$\mu_c^{(p)} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \mu_c^{(p,g)}.$$

The discrimination power of criterion c is then defined as

$$d_c = \max_{p \in \mathcal{P}} \mu_c^{(p)} - \min_{p \in \mathcal{P}} \mu_c^{(p)}.$$

To stabilize the contribution of high-variance criteria, we apply a square-root transformation $\tilde{d}_c = \sqrt{d_c}$ and normalize:

$$w_c = \begin{cases} \frac{\tilde{d}_c}{\sum_{c' \in \mathcal{C}} \tilde{d}_{c'}} & \text{if } \sum_{c' \in \mathcal{C}} \tilde{d}_{c'} > 0, \\ \frac{1}{|\mathcal{C}|} & \text{otherwise.} \end{cases}$$

The final score for a generated item y with normalized criterion scores $\{\tilde{s}_c\}_{c \in \mathcal{C}}$ is

$$S(y) = \sum_{c \in \mathcal{C}} w_c \tilde{s}_c.$$

B.4 Matched-Task Pairwise Comparison

A task slot is defined as a pair (f, d) , where f denotes the source file and d the target Bloom level. Comparisons are restricted to task slots where both pipelines produced outputs. Each comparison unit is a matched triple (f, d, g) .

For each matched triple, let $\mathcal{Y}_p^{(f,d,g)}$ denote the set of candidate outputs from pipeline p . In *raw* mode, the representative is the first generated candidate. In *production* mode, the representative is the candidate with the highest weighted score:

$$y_p^{(f,d,g)} = \arg \max_{y \in \mathcal{Y}_p^{(f,d,g)}} S(y).$$

With $S_1 = S(y_{p_1}^{(f,d,g)})$ and $S_2 = S(y_{p_2}^{(f,d,g)})$, the pairwise outcome is

$$\text{cmp}(p_1, p_2; f, d, g) = \begin{cases} \text{win}_{p_1} & \text{if } S_1 > S_2, \\ \text{win}_{p_2} & \text{if } S_2 > S_1, \\ \text{tie} & \text{otherwise.} \end{cases}$$

When weighted scores are equal, ties are resolved via a fixed lexicographic ordering over priority criteria. Outcomes are aggregated over all matched triples for each pipeline pair, and we apply bootstrap resampling over matched comparisons to report 95% confidence intervals for pairwise win rates.

C Additional Results

This section provides extended analysis of the experimental results through per-model breakdowns, outcome distributions, pairwise dominance matrices, Bloom-stratified comparisons, and score margin analysis.

High-Fidelity Compression of Course Content

A key challenge in scalable question generation is the prohibitive length and noise present in raw instructional materials. Directly prompting on full lecture notes, slides, and auxiliary documents introduces redundancy, administrative artifacts, and excessive input length that degrade both generation quality and throughput. To address this, we introduce a high-fidelity compression stage that reduces input size while preserving all pedagogically relevant content required for assessment generation.

Rather than performing conventional summarization, our approach enforces a *loss-constrained reduction* of the source material. Compression is implemented using a Gemini-3 Pro (Team et al., 2025) agent guided by a structured system prompt that explicitly distinguishes between examinable

and non-examinable content. The process retains all definitions, conceptual distinctions, procedural steps, examples, and technical terminology, while removing only administrative text, repeated explanations, and low-information filler. Importantly, the compression is conservative: original phrasing is preserved wherever possible, paraphrasing is minimized, and deduplication is applied only when semantic equivalence is clear. This ensures that the resulting representation remains faithful to the source material and suitable for downstream generation.

The compressed output is organized into coherent topical chunks, grouping related concepts, definitions, and examples. This structured representation improves prompt locality and enables more targeted question generation across Bloom levels. Additionally, lightweight standardization of mathematical expressions is applied to improve clarity without altering meaning or introducing new notation.

The compression pipeline is fully automated and includes validation to ensure structural consistency and completeness. For each document, we record the original and compressed word counts, percentage reduction, and an assigned content category. Only documents classified as *course_concept* are retained for generation, while administrative and adjacent materials are filtered out, ensuring that the generation stage operates exclusively on pedagogically relevant content.

As shown in Table 11, the proposed compression reduces total text length by 78.5% (measured in words) across the dataset, with comparable reductions for both retained and filtered subsets. Despite this substantial reduction, the preserved content remains sufficient to support high-quality question generation, demonstrating that aggressive input reduction can be achieved without compromising instructional fidelity.

Per-model win rates. Figure 3 breaks down matched-task win rates by generation model and selection mode. The proposed pipeline achieves the highest win rates across all model and setting combinations, with GPT-4o-mini showing the largest absolute win rates for SAQs in both raw and production modes. This suggests that the structured pipeline is particularly effective at compensating for the limitations of smaller models. For SBQs, the advantage is more uniform across models, with Gemini-2.5 Flash showing a slightly narrower margin, consistent with the observation that SBQ generation is inherently more variable.

Outcome distributions. Figure 4 presents the full win/tie/loss distributions averaged across models. The proposed pipeline not only achieves the highest win proportion in all settings, but also maintains the smallest loss proportion. In production mode, losses become negligible for SAQs (0.0%) and remain very low for SBQs (8.5%), confirming that candidate selection effectively eliminates the weakest outputs. Among baselines, Bloom-aware variants (ZS+B, MZS+B) show notably higher tie rates than their non-Bloom counterparts, suggesting that Bloom conditioning narrows the quality gap but does not consistently produce superior outputs.

Effect of production selection. Figure 5 isolates the effect of switching from raw to production mode. Production selection affects pipelines unevenly: for SAQs, it generally reduces win rates for weaker baselines while leaving the proposed pipeline largely unchanged, indicating that the proposed method already generates high-quality first candidates. For SBQs, production selection yields meaningful gains for the proposed pipeline and for MZS+B, with the largest improvements observed under Gemini-2.5 Flash. This asymmetry reflects the higher output variability in SBQ generation, where having multiple candidates and a quality-aware selection mechanism provides greater benefit.

Pairwise dominance. Figure 6 shows head-to-head win-rate matrices for all pipeline pairs. The proposed pipeline achieves win rates above 0.50 against every baseline in nearly all conditions, confirming broad dominance rather than gains concentrated against a single weak comparator. The matrices also reveal that among baselines, the relationships are more mixed: ZS+B and MZS+B trade wins depending on the format and selection mode, while non-Bloom variants (ZS, MZS) consistently underperform. This supports the interpretation that Bloom conditioning provides a meaningful baseline improvement, but the full proposed framework is needed for consistent superiority.

Bloom-level stratification. Figure 7 stratifies win rates by target Bloom level. The proposed pipeline shows advantages across most cognitive categories, with especially pronounced gains at the comprehension and analysis levels. For higher-order levels (evaluation and synthesis), the advantage is maintained but with smaller margins and more limited data, consistent with the known difficulty of generating and evaluating higher-order assessment items. Notably, at the Knowledge level, differences between pipelines are smaller, reflecting that factual

recall questions are comparatively straightforward for all methods.

Score margins. Figure 8 quantifies not just whether the proposed pipeline wins, but by how much. The proposed pipeline maintains positive mean score margins against all baselines in every setting. Margins are largest against ZS and MZS (the non-Bloom baselines), reflecting the combined benefit of cognitive control and structural verification. Against ZS+B and MZS+B, margins are smaller but consistently positive, indicating that the gains from Bloom specialization and verification are real but more incremental when the baseline already includes Bloom prompting. Per-model markers show that margins are relatively stable across generators, with no single model driving the aggregate advantage.

D Manual Qualitative Evaluation

To complement the automated criterion-based evaluation, we conduct a structured blind manual evaluation of matched-task sample outputs. A human evaluator assessed generated assessment items across both question formats and all difficulty levels, without knowledge of which pipeline produced which item.

D.1 Evaluation Protocol

Evaluations were conducted in two settings. *Representative evaluations* examine one matched comparison per difficulty level (easy, medium, hard) for both SAQs and SBQs, selecting the task slot that best represents the typical quality of each pipeline at that difficulty. *Illustrative contrast evaluations* select task slots specifically chosen to expose quality differences between pipelines along two axes: the overall quality gap (max_score_gap, i.e., the slot where the weighted score difference between the best and worst pipeline is greatest) and the Bloom alignment gap (bloom_bla_gap, i.e., the slot where the difference in Bloom alignment scores is greatest). All matched triples are presented to the evaluator blindly, with pipeline identities anonymized via random relabeling.

Each item is ranked against all others in its comparison set using the following criteria:

- **Bloom level alignment:** Consistency between declared and actual cognitive demand.
- **Course grounding:** Faithfulness to source material and correct use of concepts.
- **Clarity and specificity:** Unambiguous question wording with a well-defined task.

- **Single-task integrity:** The question assesses one coherent objective.
- **Objective gradability:** The mark scheme contains concrete, checkable marking points.
- **Mark scheme quality:** Coverage and logical structure of marking points.
- **Answer fidelity:** Alignment between the expected answer and the mark scheme.
- **Scenario relevance and necessity** (SBQs only): The scenario meaningfully contributes to the assessment task rather than providing merely decorative context.
- **Scenario grounding** (SBQs only): The scenario is realistic and consistent with course content.

D.2 Aggregate Results

Table 14 summarizes selection outcomes across all 18 comparisons. The proposed pipeline is ranked first in 13 of 18 settings (72.2%). Its advantage is most pronounced on Bloom-sensitive evaluations: it wins all three illustrative contrast comparisons targeting Bloom alignment gaps for SAQs, and two of three for SBQs. The only settings in which it does not rank first are the easy representative and easy illustrative contrast evaluations for SBQs, where Bloom-aware baselines benefit from the simpler generation demands of lower-difficulty items.

D.3 Representative Evaluations

In representative evaluations, one item per pipeline is drawn from the task slot most representative of each pipeline’s typical quality at a given difficulty. Rankings are assigned by the evaluator based on the full assessment triple.

SAQs. Table 15 reports the full ranking across the three SAQ difficulty levels. At easy difficulty, the proposed pipeline ranks first with a precisely scoped Knowledge-level question that achieves exact Bloom alignment and a concise, fully matched mark scheme. The key separating factor is whether the declared cognitive level honestly reflects the actual task: the two lowest-ranked baselines (ZS, MZS) label their outputs as Knowledge-level but effectively require Understanding-level reasoning, a pattern consistently penalized across all evaluation settings.

At medium difficulty, the proposed pipeline again ranks first with a genuine Application-level item that embeds a target concept in a concrete, well-bounded problem. The second-ranked baseline (MZS+B) produces a strong Comparative Analysis question, but the remaining baselines are weakened by either Bloom misalignment or overly broad

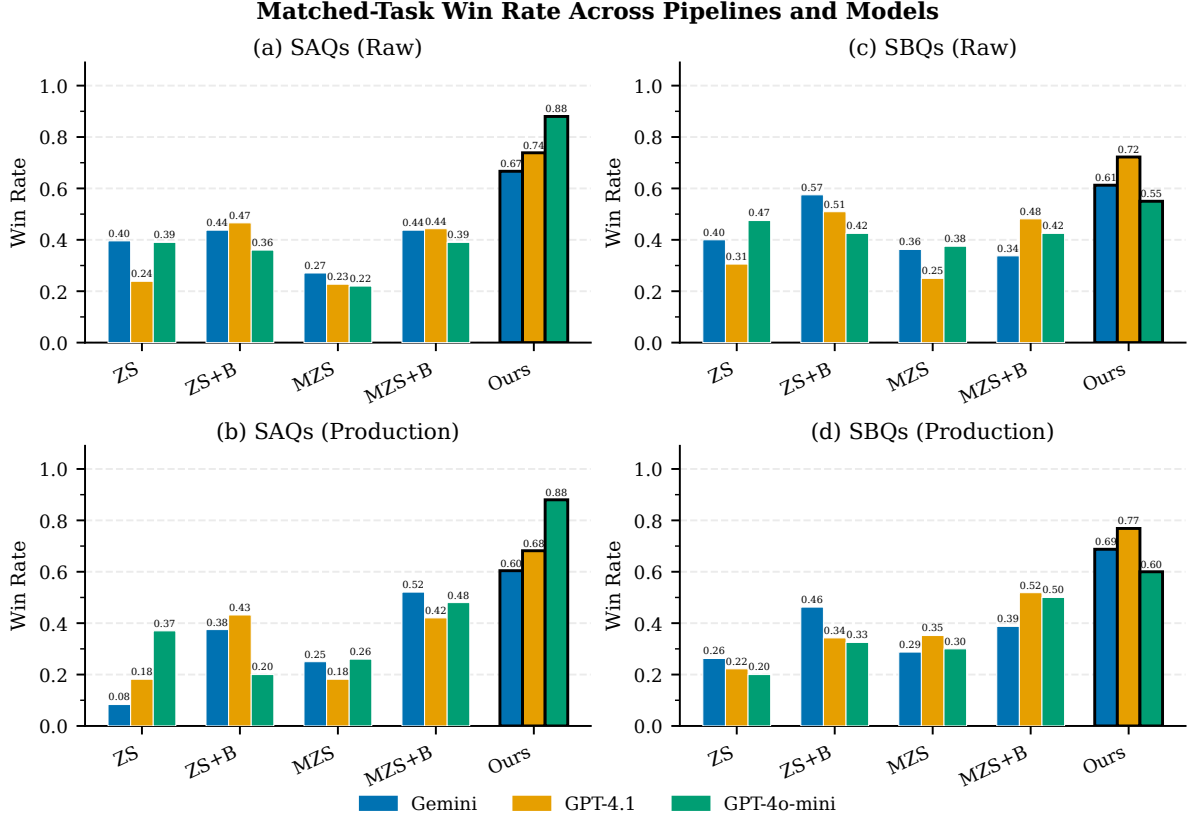


Figure 3: Matched-task win rates across pipelines, models, and selection settings for SAQs and SBQs. Tasks are matched by source content, target Bloom level, and generation model. Raw mode uses the first candidate; production mode selects the highest-scoring candidate.

question stems. At hard difficulty, MZS+B ranks first with a clean Evaluation question that requires justified decision-making under real constraints, while the proposed pipeline places second. The remaining baselines underdeliver at the Evaluation level by drifting toward Analysis or Understanding. Full annotated examples for easy, medium, and hard difficulty are shown in Figures 21, 22, and 23 respectively.

SBQs. Table 16 reports representative SBQ rankings. At easy difficulty, ZS+B ranks first with a well-grounded Understanding-level item that balances scenario relevance with mark scheme precision. The proposed pipeline ranks third: its item is structurally sound and Bloom-aligned, but the scenario is disproportionately large relative to the narrow scope of the question being asked, weakening the assessment’s use of contextual framing. MZS places last across all three difficulty levels.

At medium difficulty, the proposed pipeline ranks first with an internally consistent Analyze-level question that uses the scenario directly and necessarily to support a structured comparative Analysis task. At hard difficulty, the proposed

pipeline again ranks first with a Synthesis-level item requiring students to develop a principled guidelines framework from a complex AI ethics case study. The separating factors across all three levels are correct Bloom alignment and proportionate, necessary scenario use. Full annotated examples for easy, medium, and hard difficulty are shown in Figures 24, 25, and 26 respectively.

D.4 Illustrative Contrast Evaluations

Illustrative contrast evaluations use task slots specifically selected to expose qualitative differences between pipelines. Each difficulty level contributes one slot per axis: max_score_gap for overall quality spread and bloom_bla_gap for Bloom alignment spread.

SAQ illustrative contrasts. Tables 17 and 18 report rankings on the max_score_gap and bloom_bla_gap axes respectively. On the max_score_gap axis, the proposed pipeline ranks first in four of six settings. At easy difficulty it tops the ranking with a precisely defined Knowledge-level CSP item exhibiting exact Bloom alignment and a minimal, coherent mark scheme. The single

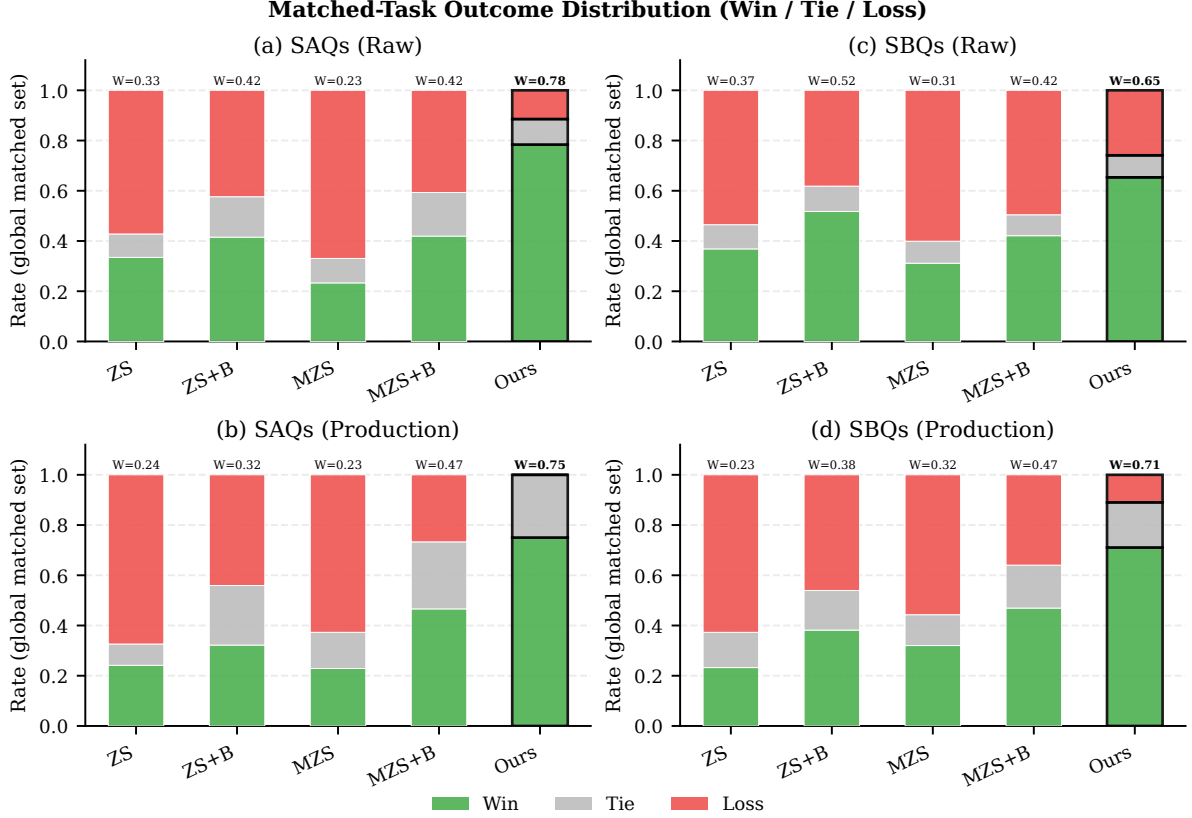


Figure 4: Stacked win/tie/loss distributions per pipeline, averaged across models, on the global matched-task set. Across SAQs and SBQs in both raw and production settings, the proposed pipeline achieves the most favourable outcome distribution, with the highest win rates and near-zero loss rates in production.

max_score_gap setting it does not win is medium difficulty, where ZS+B produces a clean comparative Analysis question judged to be more naturally framed and better pitched at that cognitive level. Full annotated examples are shown in Figures 12, 13, and 14.

On the bloom_bla_gap axis, the proposed pipeline ranks first in all three settings. In every case, the primary discriminating factor is whether the declared Bloom level accurately reflects the actual cognitive demand: lower-ranked items are penalized either for labeling Understanding-level questions as Knowledge or for inflating Evaluation labels onto questions that are more naturally Understanding or Application. The proposed pipeline’s combination of Bloom-specialized generation and explicit verification is directly responsible for this consistent advantage. Full annotated examples are shown in Figures 9, 10, and 11.

SBQ illustrative contrasts. Tables 19 and 20 report SBQ contrast rankings. On the max_score_gap axis, MZS+B ranks first in both easy settings while the proposed pipeline wins medium and hard. This pattern is consistent

with the representative results: on straightforward easy tasks where scenario grounding requirements are less demanding, Bloom-aware baselines can match or exceed the proposed pipeline. At medium difficulty the proposed pipeline tops the ranking with the most precisely controlled and objectively gradable application item, and at hard difficulty with the strongest evaluative judgment task. Full annotated examples are shown in Figures 18, 19, and 20.

On the bloom_bla_gap axis, MZS+B again ranks first at easy difficulty. At medium and hard, the proposed pipeline ranks first with items whose scenarios are essential to the reasoning being asked and whose declared cognitive level accurately matches the actual task. The recurring failure pattern in lower-ranked baselines is scenario-question mismatch: either a large elaborate scenario is provided for a question that barely uses it, or the question substantially overstates the cognitive complexity relative to the declared Bloom label. Full annotated examples are shown in Figures 15, 16, and 17.

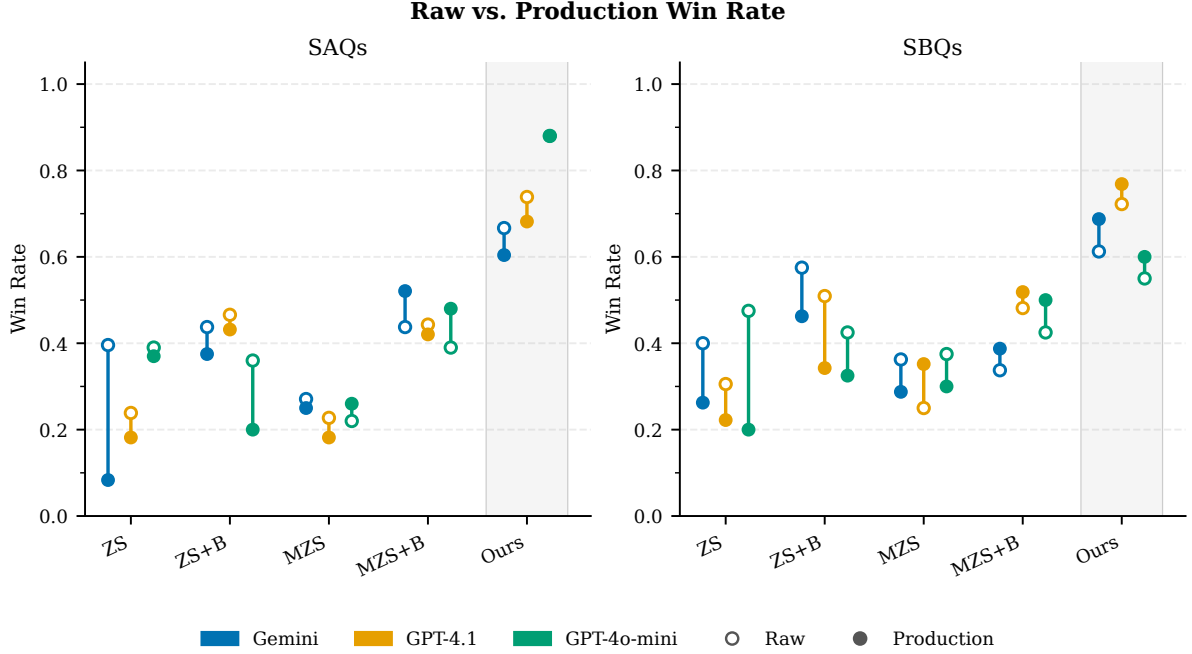


Figure 5: Per-model raw (hollow) and production (filled) win rates across pipelines. The vertical gap between markers shows the effect of production selection, which is modest for SAQs but clearer for SBQs, especially for the stronger pipelines.

D.5 Key Qualitative Observations

Several consistent patterns emerge across all evaluation settings.

Bloom alignment as primary differentiator. The single most consequential factor across all qualitative evaluations is whether the declared Bloom level accurately reflects the actual cognitive demand of the question. Items labeled as Knowledge but requiring Understanding-level reasoning, or labeled as Evaluation while only demanding Explanation, are consistently ranked below correctly aligned alternatives regardless of surface-level quality. This pattern is most salient in the bloom_bla_gap contrast evaluations, where the proposed pipeline wins all three SAQ settings and two of three SBQ settings. Lower-ranked baselines systematically exhibit one of two failure modes: cognitive understatement (e.g., labeling Analysis as Knowledge) or cognitive inflation (e.g., labeling Understanding as Evaluation), the latter being particularly common at hard difficulty.

Mark scheme coherence as separating criterion. Even when question stems are well formed, differences in mark scheme quality reliably separate pipeline ranks. The strongest items have mark schemes whose scope exactly matches the question stem, with concrete, independently verifiable marking points. Weaker items either over-extend

— including points not implied by the stem — or under-specify — bundling multiple related concepts into a single marking point — making consistent grading more difficult. This accounts for several mid-range rankings at medium and hard difficulty where the stem is strong but the mark scheme introduces uncertainty.

Scenario use in SBQs. For scenario-based questions, a recurring weakness in lower-ranked outputs is mismatch between scenario size and question scope. Some pipelines generate elaborate multi-element scenarios but then pose only a narrow recall question, leaving the scenario largely unused. Others generate questions that could be answered without the scenario at all, making the contextual framing decorative rather than functional. The proposed pipeline consistently avoids this failure mode at medium and hard difficulty, where the scenario directly motivates the analytical or evaluative task being posed.

Baseline competitiveness. MZS+B is the most consistently strong baseline, ranking first or second in seven of eighteen settings. It is particularly competitive at easy difficulty and in settings where straightforward Bloom-conditioned generation is sufficient. ZS+B is also competitive, especially at medium difficulty, where it ranks first in the SAQ max_score_gap setting with a natural, well-

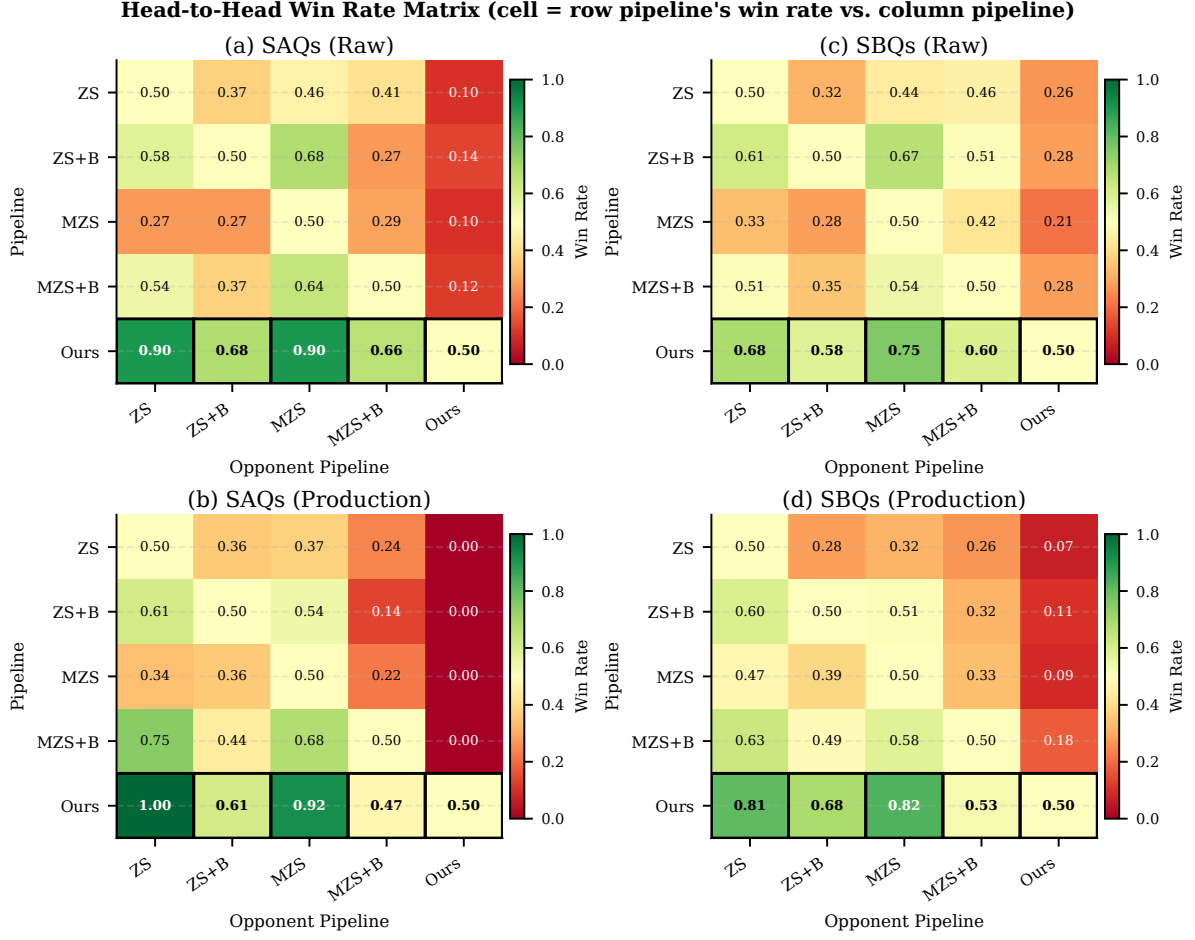


Figure 6: Head-to-head win-rate matrices for SAQs and SBQs in raw and production settings, where each cell denotes the row pipeline’s win rate against the column pipeline. The diagonal is fixed at 0.50 by definition. Across all four settings, the proposed pipeline shows the strongest overall head-to-head performance.

targeted comparative Analysis question. MZS is the weakest overall baseline and never achieves a first-place finish.

Difficulty modulates the advantage. The proposed pipeline’s advantage over baselines is most consistent at medium and hard difficulty. At easy difficulty, the generation task is less demanding and Bloom-aware baselines can occasionally match or exceed the proposed pipeline’s output quality, particularly for SBQs. As difficulty increases and the requirement for genuine higher-order reasoning becomes the central assessment objective, the structured generation and explicit verification stages of the proposed pipeline provide a more robust and consistent advantage.

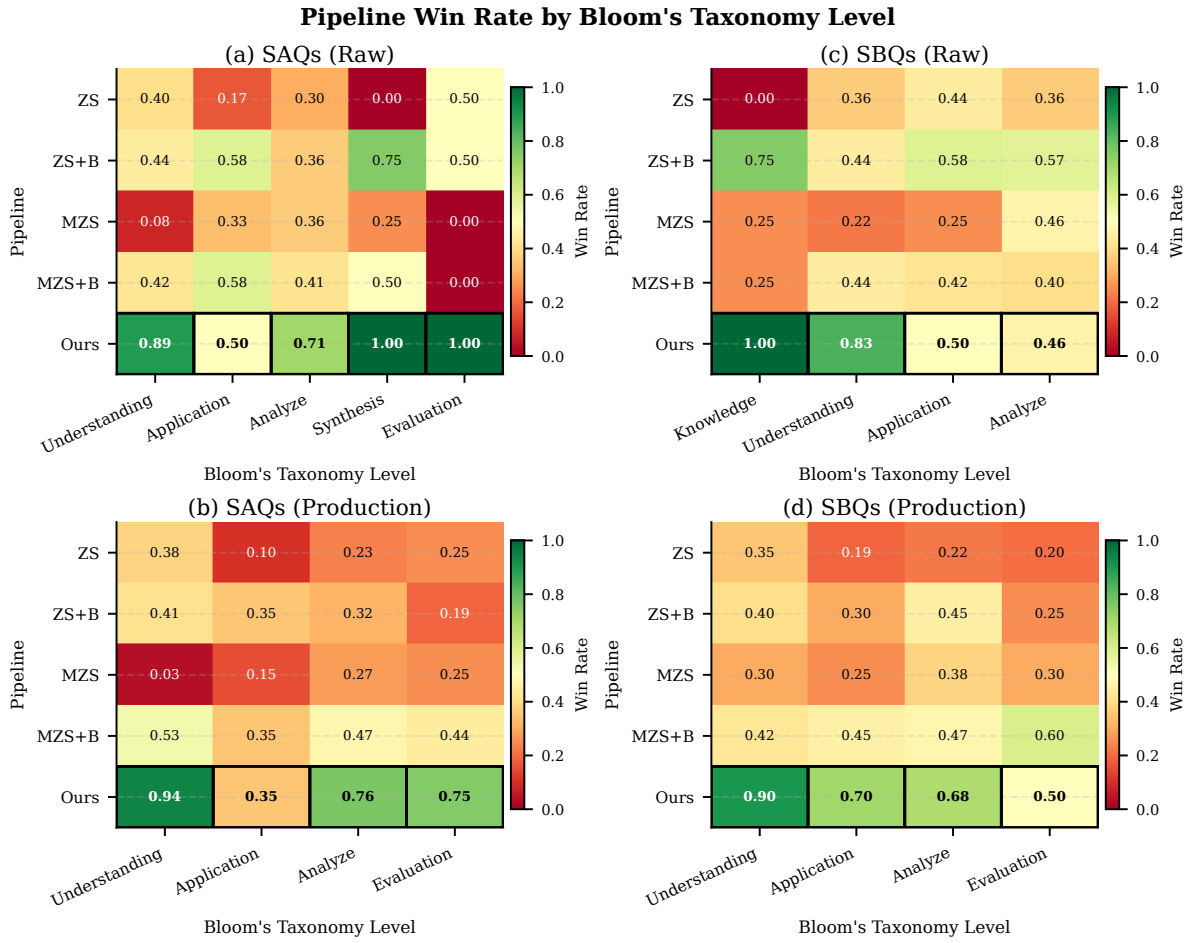


Figure 7: Win rates stratified by Bloom level for SAQs and SBQs in raw and production settings. On SAQs, the proposed pipeline generally outperforms the other pipelines across cognitive levels. On SBQs, this advantage is less uniform, with higher-order levels remaining more difficult to win consistently.

Document	Raw	Compressed	% Decrease	Category
Wk. 1 Introduction Slides	1147	984	14.21	course_adjacent
Wk. 1 Overview Text	2764	205	92.58	administrative
Wk. 2 Quiz Materials	10362	3807	63.26	course_adjacent
Wks. 2–3 Annotated Notes	3893	1777	54.35	course_concept
Wk. 2 Overview Text	2374	1011	57.41	administrative
Wk. 2 Unsupervised Learning Notes	31866	5916	81.43	course_concept
Wk. 2 Machine Learning Slides	13350	4452	66.65	course_concept
Wk. 3 Supervised Learning Notes	26857	6279	76.62	course_concept
Wk. 3 Overview Text	2009	630	68.64	administrative
Wk. 4 AI Ethics Slides	28781	5580	80.61	course_concept
Wk. 4 Ethics Case Study	3734	2461	34.09	course_concept
Wk. 4 Overview Text	1641	446	72.82	course_adjacent
Wk. 5 Overview Text	1693	279	83.52	administrative
Wk. 5 Search Slides	29687	4180	85.92	course_concept
Wk. 7 Adversarial Search Slides	17256	2974	82.77	course_concept
Wk. 7 Overview Text	879	219	75.09	course_concept
Wk. 7 Tutorial	34870	2384	93.16	course_adjacent
Wk. 7 Tutorial Solutions	3054	2675	12.41	course_concept
Wk. 8 Overview Text	1498	402	73.16	administrative
Wk. 8 CSP Slides	23479	4382	81.34	course_concept
Wk. 9 Uncertainty Slides	20163	4761	76.39	course_concept
Wk. 9 Overview Text	1811	458	74.71	administrative
Wk. 10 Lecture Slides	12138	3115	74.34	course_adjacent
Wk. 10 Overview Text	1990	453	77.24	administrative
Wk. 10 Reinforcement Learning Notes	28052	5607	80.01	course_concept
Wk. 11 Overview Text	1849	581	68.58	course_adjacent
Wk. 12 Overview Text	1006	176	82.51	administrative
Module Overview Text	4950	1033	79.13	administrative
Overall Statistics				
Total	313153	67227	78.53	all
Course Concept (13)	231051	51263	77.81	used
Non-Concept (15)	82102	15964	80.56	filtered

Table 11: Compression statistics for all course materials used in question generation. Columns report the original word count, compressed word count, percentage reduction, and assigned content category. Rows in bold correspond to *course_concept* items, which are retained for question generation, while administrative and adjacent content are filtered out. Overall, compression reduces total word count by 78.5% while preserving the majority of instructional content, enabling more efficient and scalable generation.

Pipeline	SAQ								SBQ									
	OG-T	OG-S	MSQ	AF	STI	Cla.	CG	BLA	OG-T	OG-S	MSQ	AF	SRN	SG	STI	Cla.	CG	BLA
ZS	0.993	0.900	0.450	0.964	0.957	0.979	0.957	0.007	0.912	0.772	0.493	0.949	0.110	0.632	0.816	0.941	0.956	0.169
ZS+B	0.979	0.915	0.366	0.972	0.993	0.986	0.965	0.718	0.984	0.814	0.605	0.961	0.171	0.589	0.884	0.961	0.907	0.643
MZS	0.908	0.839	0.238	0.946	0.916	0.977	0.962	0.073	0.959	0.728	0.500	0.967	0.098	0.553	0.882	0.935	0.931	0.142
MZS+B	0.864	0.921	0.360	0.925	0.982	0.996	0.952	0.667	0.972	0.743	0.607	0.977	0.196	0.645	0.925	0.967	0.935	0.561
Ours	0.988	0.926	0.885	0.990	0.992	0.992	0.987	0.917	0.871	0.751	0.747	0.885	0.137	0.879	0.960	0.887	0.852	0.897

Table 12: Per-criterion pass rates across pipelines for both SAQs and SBQs. SAQ criteria are OG-T = objective gradability (triple), OG-S = objective gradability (stem), MSQ = mark scheme quality, AF = answer fidelity, STI = single-task integrity, Cla. = clarity, CG = course grounding, and BLA = Bloom alignment accuracy. SBQs additionally include SRN = scenario relevance/necessity and SG = scenario grounding.

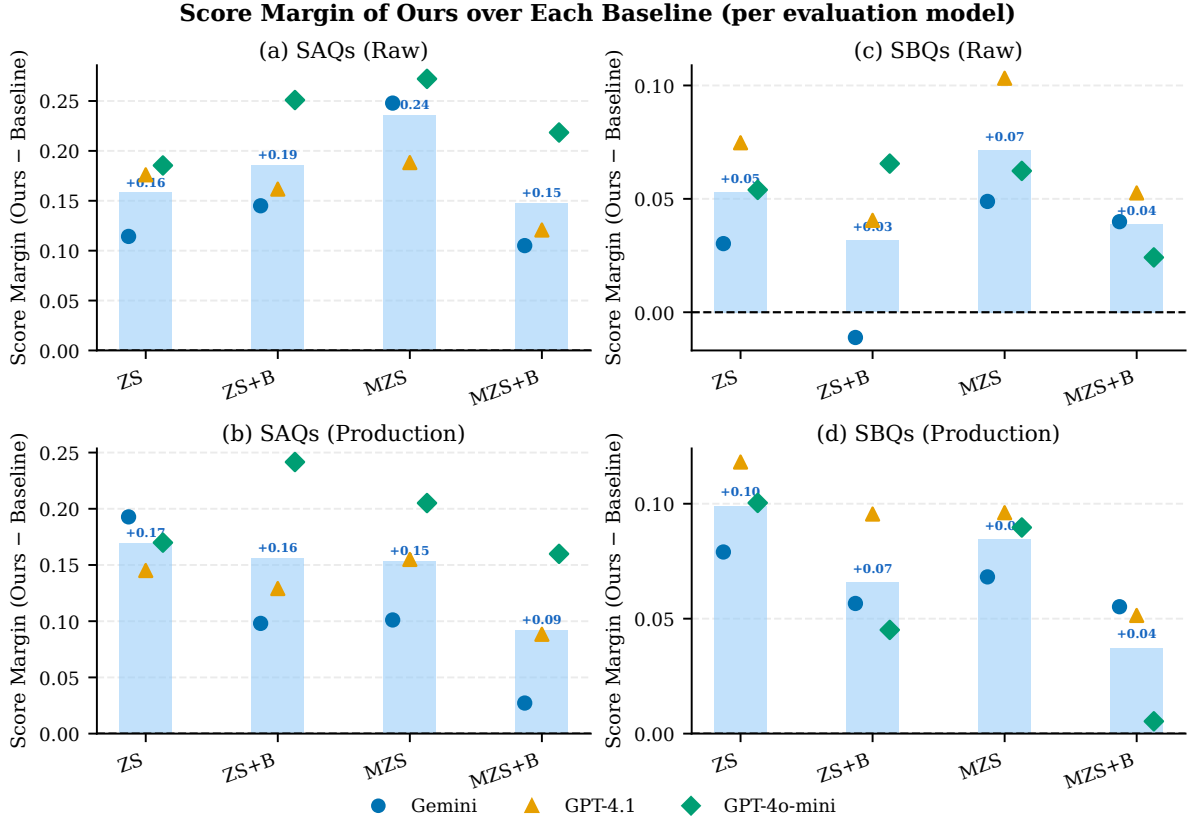


Figure 8: Mean score margin of the proposed pipeline relative to each baseline, shown separately for SAQs and SBQs in raw and production settings. Bars denote averages across evaluation models, while markers indicate per-model margins. The proposed pipeline maintains positive score margins against all baselines in every setting, with larger gains on SAQs than on SBQs.

Criterion	SAQ Weight	SBQ Weight
Mark Scheme Quality	0.326	0.160
Bloom Alignment	0.197	0.137
Obj. Gradability (Triple)	0.110	0.090
Answer Fidelity	0.093	0.089
Obj. Gradability (Stem)	0.089	0.084
Single Task Integrity	0.084	0.077
Course Grounding	0.058	0.091
Clarity	0.041	0.078
Scenario Relevance	0.000	0.070
Scenario Grounding	0.000	0.123

Pipeline	SAQ Yield	SBQ Yield
ZS	140	136
ZS+B	142	129
MZS	261	246
MZS+B	228	214
Ours	1072	750

Table 13: Left: Discrimination-based criterion weights for evaluation. Higher values indicate greater contribution to differentiating pipeline performance; scenario criteria receive zero weight for SAQs. Right: Number of valid generated assessment items (yield) per pipeline.

Illustrative Contrast SAQ Sample — Easy Difficulty (Bloom Alignment Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 10 — Reinforcement Learning (Lecture Notes with Solutions) | Model: GPT-4o-mini | Task type: Short Answer Question (SAQ) | Difficulty target: Easy

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS)	Explain the significance of the single-state nature of the k-armed bandit problem compared to a Markov Decision Process (MDP) and how this difference affects the estimation of action values.	Knowledge	Analyze [†]	6	0.882
B (OURS)	Identify the method that selects actions uniformly at random at time t .	Knowledge	Knowledge	1	1.000
C (ZS)	Explain the concept of action-value estimation in k-armed bandit problems and why it is essential for decision-making.	Knowledge	Understanding [†]	5	0.961
D (ZS+B)	Describe how action-value estimates $Q_t(a)$ are computed in the k-armed bandit problem and explain why these estimates are necessary.	Knowledge	Understanding [†]	6	0.634
E (MZS+B)	Explain the significance of maintaining estimates of average rewards $Q_t(a)$ in the k-armed bandit problem and describe how these estimates are updated over time.	Understanding	Understanding	9	0.674

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
The Random Method selects actions uniformly at random at time t .	1
Model answer: The Random Method selects actions uniformly at random at time t .	/1

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (rank 1):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MZS (rank 5):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.4

Selection Justification

Automated selection (Bloom alignment gap axis): task selected from easy difficulty pool to highlight a large Bloom alignment discrepancy across pipelines. **Best σ_w :** 1.000 **Bloom gap:** 0.6000 **Gap percentile rank:** 42.9th **Runner-up gap:** 0.6000 **Mean σ_w :** 0.8300

Blind human evaluation ranking: OURS > ZS > MZS+B > ZS+B > MZS. OURS was ranked first because it is the cleanest example of correct cognitive targeting: the declared and true Bloom levels match, the item is fully objective, and the expected answer is exact. MZS was ranked last because it has the largest and most consequential Bloom mismatch: it is declared as knowledge but actually asks for analytical comparison between bandits and MDPs and their implications for value estimation.

Figure 9: Illustrative contrast SAQ sample, easy difficulty tier (Bloom alignment gap axis). Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SAQ Sample — Medium Difficulty (Bloom Alignment Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2 — Introduction to Machine Learning (Slides) | Model: Gemini-2.5-Flash | Task type: Short Answer Question (SAQ) | Difficulty target: Medium

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS+B)	Describe the core concept of machine learning by synthesizing the different definitions provided, highlighting common themes or key aspects.	Understanding	Synthesis [†]	8	0.679
B (OURS)	You are building a machine learning model to identify spam emails. Given a large dataset of emails, show how you would establish training and test sets to properly evaluate your model's performance on emails it has not seen before.	Application	Application	8	1.000
C (ZS)	Explain why accuracy alone can be an insufficient metric for evaluating machine learning models, particularly when dealing with imbalanced datasets. Provide a concise example.	Application	Understanding [†]	8	0.634
D (MZS)	Explain the distinction between classification and regression problems, providing an example for each to illustrate their different objectives within supervised learning.	Application	Understanding [†]	9	0.634
E (ZS+B)	Describe the concepts of underfitting and overfitting in machine learning, and explain how selecting the appropriate model complexity helps mitigate these issues.	Understanding	Understanding	9	0.674

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Divide the large dataset of labeled emails into two distinct sets: a training set and a test set.	2
Utilize the training set, which includes both email content and their “spam”/“not spam” labels, to teach the machine learning model to identify patterns.	1
Keep the test set separate and completely untouched during the entire training process to ensure it represents genuinely unseen data.	2
After the model is trained, use the email content from the test set to make predictions without providing the true labels to the model.	1
Compare the model's predictions on the test set against the actual (previously hidden) labels to accurately evaluate its generalization performance on new, unseen emails.	2

Model answer: To establish training and test sets, the large dataset of labeled emails is divided into two distinct sets: a training set and a test set. The training set, which includes email content and their “spam”/“not spam” labels, is utilized to teach the machine learning model to identify patterns. The test set is kept separate and completely untouched during the entire training process to ensure it represents genuinely unseen data. After the model is trained, email content from the test set is used to make predictions without providing the true labels to the model. The model's predictions on the test set are then compared against the actual (previously hidden) labels to accurately evaluate its generalization performance on new, unseen emails. /8

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (rank 1):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MZS+B (rank 5):							
1.0	1.0	1.0	1.0	1.0	0.5	1.0	0.2

Selection Justification

Automated selection (Bloom alignment gap axis): task selected from medium difficulty pool to maximize a large Bloom alignment discrepancy across pipelines. **Best** σ_w : 1.000 **Bloom gap**: 0.8000 **Gap percentile rank**: 97.1th **Runner-up gap**: 0.4000 **Mean** σ_w : 0.7242

Blind human evaluation ranking: OURS > ZS+B > MZS > ZS > MZS+B. OURS was ranked first because it is the clearest example of genuine application: it uses a realistic scenario, preserves single-task integrity, and aligns perfectly with its declared Bloom level. MZS+B was ranked last because it has the largest Bloom mismatch in the set: the task explicitly asks students to synthesize multiple definitions of machine learning, but it is labelled only as understanding.

Figure 10: **Illustrative contrast SAQ sample, medium difficulty tier (Bloom alignment gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SAQ Sample — Hard Difficulty (Bloom Alignment Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 7 — Heuristic Search & Adversarial Search (Slides) | Model: GPT-4o-mini | Task type: Short Answer Question (SAQ) | Difficulty target: Hard

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS)	Explain why state space complexity varies significantly among games like Tic-Tac-Toe, Chess, and Go, and discuss the implications of this variation for game search algorithms.	Evaluation	Analyze [†]	9	0.961
B (MZS+B)	Evaluate the conditions under which the minimax algorithm is guaranteed to be optimal and complete, and explain how these conditions relate to the nature of the opponent's play.	Evaluation	Evaluation	9	1.000
C (MZS)	Why is the minimax algorithm considered complete and optimal, and under what conditions might these properties fail in practical game playing?	Evaluation	Analyze [†]	9	0.797
D (OURS)	Evaluate whether the Minimax algorithm is a practical solution for games with extremely large state spaces such as Chess or Go, using the details about state space complexity and Minimax properties provided in the text.	Evaluation	Evaluation	9	1.000
E (ZS+B)	Discuss how Alpha-Beta pruning optimizes the Minimax algorithm and explain the roles of alpha and beta in the pruning process.	Evaluate	Understanding [†]	10	0.500

[†]Declared-true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Minimax explores the entire game tree with time complexity $O(b^d)$, which is infeasible for very large state spaces like Chess or Go due to enormous branching factors and depths.	2
The enormous size of Chess (8.73×10^{79} states) and Go (2×10^{170} states) makes exhaustive Minimax search practically impossible within reasonable time or space.	2
While Minimax is complete and optimal for finite trees with optimal opponents, these properties do not offset its impracticality at huge state scale.	1
Alpha-Beta pruning can reduce the number of states evaluated but still cannot make Minimax fully practical for extremely large games due to exponential growth of the tree.	2
Therefore, Minimax alone is not a practical solution for Chess or Go without significant heuristic improvements or approximations.	2
<i>Model answer:</i> Minimax explores the entire game tree with time complexity $O(b^d)$, which is infeasible for very large state spaces like Chess or Go due to enormous branching factors and depths. The enormous size of Chess (8.73×10^{79} states) and Go (2×10^{170} states) makes exhaustive Minimax search practically impossible within reasonable time or space. While Minimax is complete and optimal for finite trees with optimal opponents, these properties do not offset its impracticality at huge state scale. Alpha-Beta pruning can reduce the number of states evaluated but still cannot make Minimax fully practical for extremely large games due to exponential growth of the tree. Therefore, Minimax alone is not a practical solution for Chess or Go without significant heuristic improvements or approximations.	/9

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (rank 1): 1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ZS+B (rank 5): 1.0	1.0	1.0	1.0	0.5	0.0	1.0	0.4

Selection Justification

Automated selection (Bloom alignment gap axis): task selected from hard difficulty pool to highlight a large Bloom alignment discrepancy across pipelines. **Best σ_w :** 1.000 **Bloom gap:** 0.6000 **Gap percentile rank:** 52.9th **Runner-up gap:** 0.6000 **Mean σ_w :** 0.8516

Blind human evaluation ranking: OURS > MZS+B > ZS > MZS > ZS+B. OURS was ranked first because it is the most naturally evaluative item and aligns perfectly with its declared Bloom level: it asks for a justified practicality judgment under explicit state-space constraints. **ZS+B** was ranked last because it is the clearest case of a question labeled as evaluation while actually requiring only understanding of alpha-beta pruning and the roles of alpha and beta.

Figure 11: **Illustrative contrast SAQ sample, hard difficulty tier (Bloom alignment gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SAQ Sample — Easy Difficulty (Max Score Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 8 — Planning & CSPs (Lecture Slides) | Model: GPT-4.1 | Task type: Short Answer Question (SAQ) | Difficulty target: Easy

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS)	Explain the three main components of a Constraint Satisfaction Problem (CSP) and describe how these are represented in the map colouring and n-Queens examples.	Knowledge	Understanding [†]	6	0.634
B (ZS+B)	Describe the structure of a Constraint Satisfaction Problem (CSP), using the map colouring example to illustrate your explanation.	Understanding	Understanding	6	0.674
C (MZS+B)	Explain the three main components that define a Constraint Satisfaction Problem (CSP), and illustrate how each component is specified in the example of map colouring for Australia.	Understanding	Understanding	10	0.674
D (OURS)	Define a Constraint Satisfaction Problem (CSP).	Knowledge	Knowledge	3	1.000
E (MZS)	Explain the purpose of heuristics such as the Minimum Remaining Values (MRV) and Least Constraining Value in the context of CSP solving.	Knowledge	Understanding [†]	8	0.590

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
A CSP consists of a finite set of variables.	1
Each variable has a domain of possible values.	1
A set of constraints specifies allowable combinations of variable values.	1
<i>Model answer:</i> A CSP consists of a finite set of variables, each variable has a domain of possible values, and a set of constraints specifies allowable combinations of variable values.	/3

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (<i>rank 1</i>): 1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MZS (<i>rank 5</i>): 1.0	1.0	1.0	0.5	1.0	0.0	1.0	0.8

Selection Justification

Automated selection (max score gap axis): task selected from easy difficulty pool to maximise the spread between the highest- and lowest-scoring pipelines on the discrimination-weighted criterion score σ_w . **Best** σ_w : 1.000 **Worst** σ_w : 0.5895 **Score gap**: 0.4105 **Gap percentile rank**: 85.7th **Runner-up gap**: 0.3658 **Mean** σ_w : 0.7142

Blind human evaluation ranking: OURS > ZS+B > MZS+B > ZS > MZS. OURS was ranked first because it is the most internally consistent triple: exact Bloom alignment, highly objective grading, clear wording, and full answer-to-scheme correspondence. MZS was ranked last because it is Bloom-misaligned and less tightly constrained as an easy item: asking for the *purpose* of MRV and Least Constraining Value requires explanation rather than simple recall.

Figure 12: **Illustrative contrast SAQ sample, easy difficulty tier (max score gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SAQ Sample — Medium Difficulty (Max Score Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 7 — Heuristic Search & Adversarial Search (Text) | Model: GPT-4.1
 | Task type: Short Answer Question (SAQ) | Difficulty target: Medium

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (OURS)	Apply the concept of heuristics to inform the strategy for an adversarial search in a board game situation.	Application	Application	6	0.900
B (ZS+B)	Compare ordinary search and adversarial search, highlighting the key differences between the two approaches.	Analyze	Analyze	8	0.833
C (ZS)	Explain how relaxations are used to derive heuristics in search, and describe why heuristics are valuable in guiding the search process.	Application	Understanding [†]	7	0.425
D (MZS)	Explain how relaxations can be used to derive heuristics in the context of guiding search algorithms.	Application	Understanding [†]	9	0.447
E (MZS+B)	Explain how relaxations can be used to derive heuristics for guiding search processes.	Application	Understanding [†]	9	0.467

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (ZS+B — rank 1):

Marking point	Marks
Ordinary search involves finding a solution without opposition.	1
Adversarial search considers the presence of an opponent making counter-moves.	2
Key difference lies in the need to anticipate and respond to adversary strategies in adversarial search.	2
Ordinary search focuses solely on optimization or reaching a goal.	1
Adversarial search requires the evaluation of possible outcomes based on both parties' actions.	2
<i>Model answer:</i> Ordinary search focuses on finding a solution or optimizing a path to reach a goal, without any opposition. In contrast, adversarial search involves anticipating and responding to an opponent's actions, requiring strategies that account for counter-moves. The key difference is that adversarial search must evaluate possible outcomes based on the choices of both parties, while ordinary search simply optimizes or searches for a goal without considering adversarial interactions.	/8

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
ZS+B (rank 1): 0.5	1.0	1.0	0.5	1.0	1.0	0.0	1.0
ZS (rank 5): 0.5	1.0	0.5	0.5	1.0	0.0	0.0	0.8

Selection Justification

Automated selection (max score gap axis): task selected from medium difficulty pool to maximise the spread between the highest- and lowest-scoring pipelines on the discrimination-weighted criterion score σ_w . **Best** σ_w : 0.9001 **Worst** σ_w : 0.4253 **Score gap**: 0.4748 **Gap percentile rank**: 97.1th **Runner-up gap**: 0.4572 **Mean** σ_w : 0.6145

Blind human evaluation ranking: ZS+B > OURS > MZS+B > MZS > ZS. ZS+B was ranked first because it offers the most natural and properly targeted medium-level task: a clear comparative analysis question with good course grounding and a coherent mark scheme. ZS was ranked last because it is both Bloom-misaligned and less focused as a single assessment objective, combining explanation of relaxations with a broader discussion of why heuristics are useful.

Figure 13: **Illustrative contrast SAQ sample, medium difficulty tier (max score gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SAQ Sample — Hard Difficulty (Max Score Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 7 — Heuristic Search & Adversarial Search (Slides) | Model: GPT-4o-mini | Task type: Short Answer Question (SAQ) | Difficulty target: Hard

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS+B)	Discuss how Alpha-Beta pruning optimizes the Minimax algorithm and explain the roles of alpha and beta in the pruning process.	Evaluation	Understanding [†]	10	0.500
B (OURS)	Evaluate whether the Minimax algorithm is a practical solution for games with extremely large state spaces such as Chess or Go, using the details about state space complexity and Minimax properties provided in the text.	Evaluation	Evaluation	9	1.000
C (MZS)	Why is the minimax algorithm considered complete and optimal, and under what conditions might these properties fail in practical game playing?	Evaluation	Analyze [†]	9	0.797
D (MZS+B)	Evaluate the conditions under which the minimax algorithm is guaranteed to be optimal and complete, and explain how these conditions relate to the nature of the opponent's play.	Evaluation	Evaluation	9	1.000
E (ZS)	Explain why state space complexity varies significantly among games like Tic-Tac-Toe, Chess, and Go, and discuss the implications of this variation for game search algorithms.	Evaluation	Analyze [†]	9	0.961

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Minimax explores the entire game tree with time complexity $O(b^d)$, which is infeasible for very large state spaces like Chess or Go due to enormous branching factors and depths.	2
The enormous size of Chess (8.73×10^{79} states) and Go (2×10^{170} states) makes exhaustive Minimax search practically impossible within reasonable time or space.	2
While Minimax is complete and optimal for finite trees with optimal opponents, these properties do not offset its impracticality at huge state scale.	1
Alpha-Beta pruning can reduce the number of states evaluated but still cannot make Minimax fully practical for extremely large games due to exponential growth of the tree.	2
Therefore, Minimax alone is not a practical solution for Chess or Go without significant heuristic improvements or approximations.	2
<i>Model answer:</i> Minimax explores the entire game tree with time complexity $O(b^d)$, which is infeasible for very large state spaces like Chess or Go due to enormous branching factors and depths. The enormous size of Chess (8.73×10^{79} states) and Go (2×10^{170} states) makes exhaustive Minimax search practically impossible within reasonable time or space. While Minimax is complete and optimal for finite trees with optimal opponents, these properties do not offset its impracticality at huge state scale. Alpha-Beta pruning can reduce the number of states evaluated but still cannot make Minimax fully practical for extremely large games due to exponential growth of the tree. Therefore, Minimax alone is not a practical solution for Chess or Go without significant heuristic improvements or approximations.	/9

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (rank 1):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ZS+B (rank 5):							
1.0	1.0	1.0	1.0	0.5	0.0	1.0	0.4

Selection Justification

Automated selection (max score gap axis): task selected from hard difficulty pool to maximise the spread between the highest- and lowest-scoring pipelines on the discrimination-weighted criterion score σ_w . **Best σ_w :** 1.000 **Worst σ_w :** 0.5000 **Score gap:** 0.5000 **Gap percentile rank:** 94.1th **Runner-up gap:** 0.4448 **Mean σ_w :** 0.8516

Blind human evaluation ranking: OURS > MZS+B > ZS > MZS > ZS+B. OURS was ranked first because it is the strongest example of a true hard-level evaluative question: it requires a clear judgment on practicality, is strongly grounded in course content, and has an excellent mark-scheme structure. ZS+B was ranked last because it is fundamentally an explanatory understanding task mislabeled as evaluation: the mark scheme rewards description of alpha-beta pruning rather than genuine critique or judgment.

Figure 14: **Illustrative contrast SAQ sample, hard difficulty tier (max score gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SBQ Sample — Easy Difficulty (Bloom Alignment Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2–3 — Introduction to Machine Learning & Unsupervised Learning (Annotated Notes) | Model: GPT-4.1 | Task type: Scenario-Based Question (SBQ) | Difficulty target: Easy

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS)	If the algorithm always predicts “no disease,” calculate its accuracy, and discuss whether this is a good way to judge the algorithm’s performance in this scenario.	Knowledge	Analyze [†]	5	0.685
B (ZS+B)	Define the metrics MAE, MSE, and RMSE in the context of regression, and explain how each metric measures prediction errors.	Understanding	Understanding	5	0.965
C (ZS)	Which terms describe these two parts of the dataset, and why is it important to keep the evaluation set labels hidden during training?	Knowledge	Analyze [†]	4	0.685
D (OURS)	Identify the two data splits used for model development in the scenario.	Knowledge	Knowledge	2	0.965
E (MZS+B)	Describe the roles of the training set and the test set in supervised learning.	Knowledge	Knowledge	3	0.965

[†]Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (MZS+B — rank 1):

Marking point	Marks
The training set is used to fit or train the model by providing both input data and their corresponding labels.	1
The test set is used to evaluate the trained model’s performance by using input data with hidden labels during prediction.	1
The test set helps assess how well the model generalizes to unseen data.	1
<i>Model answer:</i> In supervised learning, the training set serves to train the model by providing input data along with their corresponding labels, allowing the model to learn the relationship between features and outcomes. After training, the test set is used to evaluate the model’s performance by making predictions on input data for which the labels are hidden; this process assesses how well the model generalizes to new, unseen data.	/3

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
MZS+B (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
MZS (rank 5):									
1.0	1.0	0.5	1.0	1.0	0.0	1.0	0.5	1.0	0.4

Selection Justification

Automated selection (Bloom alignment gap axis): task selected from easy difficulty pool ($n=11$ candidates) to highlight a large gap in Bloom alignment quality under otherwise matched generation conditions. **Best** σ_w : 0.9648 **Bloom gap**: 0.6000 **Gap percentile rank**: 72.7th **Runner-up gap**: 0.6000 **Mean** σ_w : 0.8527

Blind human evaluation ranking: MZS+B > ZS+B > OURS > ZS > MZS. MZS+B was ranked first because it most cleanly matches its declared knowledge-level target while remaining clear, grounded, and objective; it asks for the roles of the training and test sets and uses the scenario proportionately. MZS was ranked last because it has the clearest Bloom mismatch in the set: it is labelled as knowledge, but calculating accuracy and judging its appropriateness in an imbalanced setting requires analysis.

Figure 15: **Illustrative contrast SBQ sample, easy difficulty tier (Bloom alignment gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SBQ Sample — Medium Difficulty (Bloom Alignment Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2 — Introduction to Machine Learning (Slides) | Model: GPT-4.1 | Task type: Scenario-Based Question (SBQ) | Difficulty target: Medium

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS)	Explain what underfitting and overfitting mean in the context of this scenario, and identify which degrees correspond to each phenomenon.	Application	Understanding [†]	6	0.778
B (MZS)	Explain why the algorithm's high accuracy does not necessarily indicate it is a reliable or effective diagnostic tool in this context.	Application	Understanding [†]	7	0.716
C (MZS+B)	Explain why the algorithm's high accuracy does not necessarily indicate good performance in this context, and suggest a more appropriate evaluation metric.	Understanding	Evaluation [†]	6	0.779
D (OURS)	Compare the training and validation MSE values for polynomial degrees 1, 4, and 29, and analyze the pattern that emerges as the model complexity increases.	Analyze	Analyze	6	0.965
E (ZS+B)	Using the concepts of underfitting and overfitting, explain why the models of degree 1 and 29 perform poorly on validation data, and why the degree 4 model is preferable.	Application	Understanding [†]	6	0.778

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
For degree 1, training MSE is 0.189 and validation MSE is 0.379, indicating moderate fit.	1
For degree 4, training MSE drops to 0.00691 and validation MSE to 0.0120, showing improved fit and generalization.	2
For degree 29, training MSE is extremely low at 0.00322, but validation MSE spikes to 1.76×10^{15} , signaling severe overfitting.	2
As model complexity increases, training error decreases steadily but validation error decreases then dramatically increases, revealing the overfitting pattern.	1
<i>Model answer:</i> For degree 1, training MSE is 0.189 and validation MSE is 0.379, indicating moderate fit. For degree 4, training MSE drops to 0.00691 and validation MSE to 0.0120, showing improved fit and generalization. For degree 29, training MSE is extremely low at 0.00322, but validation MSE spikes to 1.76×10^{15} , signaling severe overfitting. As model complexity increases, training error decreases steadily but validation error decreases then dramatically increases, revealing the overfitting pattern.	/6

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
OURS (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
MZS+B (rank 5):									
1.0	1.0	1.0	0.5	1.0	1.0	1.0	0.5	0.5	0.4

Selection Justification

Automated selection (Bloom alignment gap axis): task selected from medium difficulty pool to maximize a large Bloom alignment discrepancy across pipelines. **Best** σ_w : 0.9648 **Bloom gap**: 0.6000 **Gap percentile rank**: 96.7th **Runner-up gap**: 0.4000 **Mean** σ_w : 0.8032

Blind human evaluation ranking: OURS > ZS+B > ZS > MZS > MZS+B. OURS was ranked first because it most cleanly matches its declared analyze-level target and uses the scenario values directly to support genuine comparison and trend interpretation. MZS+B was ranked last because it is declared as understanding but actually asks for a more evaluative judgment by requiring students to suggest a better metric for the scenario.

Figure 16: Illustrative contrast SBQ sample, medium difficulty tier (Bloom alignment gap axis). Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SBQ Sample — Hard Difficulty (Bloom Alignment Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 9 — MDPs / Uncertainty (Slides) | Model: GPT-4.1 | Task type: Scenario-Based Question (SBQ) | Difficulty target: Hard

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS+B)	Design a suitable reward function for the robot's navigation MDP. Briefly justify your choices for how rewards should be assigned to regular, goal, and hazardous states.	Synthesis	Synthesis	5	0.702
B (OURS)	Assess the effectiveness of modeling the warehouse inventory management system as a Markov Decision Process, given the described state space, actions, and transition uncertainties.	Evaluation	Evaluation	8	0.965
C (MZS+B)	Discuss how the reward function influences the policy derived from value iteration in such an uncertain inventory system. What might be the consequences of missing or poorly designed reward components?	Evaluation	Analyze [†]	8	0.938
D (MZS)	Explain the process for updating the utility value of state (2, 1) using the Bellman equation and discuss what the value 0.655 represents in this context.	Evaluation	Understanding [†]	8	0.723
E (ZS)	Apply the Bellman equation to compute the updated utility $U((2, 1))$ for moving left, showing your calculation, and determine which action yields the maximum expected utility.	Evaluation	Application [†]	6	0.910

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
MDP formalism allows systematic modeling of sequential decisions, capturing inventory dynamics through defined states and actions.	1
Probabilistic state transitions in the MDP match real-world uncertainties such as customer demand variability and delivery timing.	2
The MDP reward structure effectively incorporates holding costs, ordering costs, and expected profits, supporting quantitative evaluation of policies.	1
The large state space may increase computational complexity, potentially limiting tractability for optimal policy computation.	2
MDP modeling is suitable for inventory management when transition probabilities and rewards are well-defined, but performance hinges on accurate probability estimation and scalable solution methods.	2
<i>Model answer:</i> MDP formalism allows systematic modeling of sequential decisions, capturing inventory dynamics through defined states and actions. Probabilistic state transitions in the MDP match real-world uncertainties such as customer demand variability and delivery timing. The MDP reward structure effectively incorporates holding costs, ordering costs, and expected profits, supporting quantitative evaluation of policies. The large state space may increase computational complexity, potentially limiting tractability for optimal policy computation. MDP modeling is suitable for inventory management when transition probabilities and rewards are well-defined, but performance hinges on accurate probability estimation and scalable solution methods.	/8

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
OURS (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
MZS (rank 5):									
1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.5	1.0	0.4

Selection Justification

Automated selection (Bloom alignment gap axis): task selected from hard difficulty pool to highlight a large gap in Bloom alignment quality under otherwise matched generation conditions. **Best** σ_w : 0.9648 **Bloom gap**: 0.6000 **Gap percentile rank**: 81.2th **Runner-up gap**: 0.6000 **Mean** σ_w : 0.8474

Blind human evaluation ranking: OURS > ZS+B > MZS+B > ZS > MZS. OURS was ranked first because it is the clearest example of exact Bloom targeting in the set: the task is genuinely evaluative, the scenario is necessary, and the mark scheme supports a justified conclusion on whether the MDP formulation is suitable. MZS was ranked last because it has the largest and clearest Bloom mismatch: explaining a Bellman update and interpreting a value is an understanding task, not an evaluation task.

Figure 17: **Illustrative contrast SBQ sample, hard difficulty tier (Bloom alignment gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SBQ Sample — Easy Difficulty (Max Score Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2–3 — Introduction to Machine Learning & Unsupervised Learning (Annotated Notes) | Model: GPT-4.1 | Task type: Scenario-Based Question (SBQ) | Difficulty target: Easy

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS)	If the algorithm always predicts “no disease,” calculate its accuracy, and discuss whether this is a good way to judge the algorithm’s performance in this scenario.	Knowledge	Analyze [†]	5	0.685
B (ZS)	Which terms describe these two parts of the dataset, and why is it important to keep the evaluation set labels hidden during training?	Knowledge	Analyze [†]	4	0.685
C (MZS+B)	Describe the roles of the training set and the test set in supervised learning.	Knowledge	Knowledge	3	0.965
D (ZS+B)	Define the metrics MAE, MSE, and RMSE in the context of regression, and explain how each metric measures prediction errors.	Understanding	Understanding	5	0.965
E (OURS)	Identify the two data splits used for model development in the scenario.	Knowledge	Knowledge	2	0.965

[†]Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (MZS+B — rank 1):

Marking point	Marks
The training set is used to fit or train the model by providing both input data and their corresponding labels.	1
The test set is used to evaluate the trained model’s performance by using input data with hidden labels during prediction.	1
The test set helps assess how well the model generalizes to unseen data.	1
<i>Model answer:</i> In supervised learning, the training set serves to train the model by providing input data along with their corresponding labels, allowing the model to learn the relationship between features and outcomes. After training, the test set is used to evaluate the model’s performance by making predictions on input data for which the labels are hidden; this process assesses how well the model generalizes to new, unseen data.	/3

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
MZS+B (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
MZS (rank 5):									
1.0	1.0	0.5	1.0	1.0	0.0	1.0	0.5	1.0	0.4

Selection Justification

Automated selection (max score gap axis): task selected from easy difficulty pool ($n=11$ candidates) to maximise the spread between the highest- and lowest-scoring pipelines on the discrimination-weighted criterion score σ_w . **Best σ_w :** 0.9648 **Worst σ_w :** 0.6845 **Score gap:** 0.2803 **Gap percentile rank:** 90.9th **Runner-up gap:** 0.2416 **Mean σ_w :** 0.8527

Blind human evaluation ranking: MZS+B > ZS+B > OURS > ZS > MZS. MZS+B was ranked first because it is the cleanest and most internally consistent SBQ in the set: the question is a straightforward knowledge-level task, the declared and true Bloom levels match exactly, and the mark scheme is concise, objective, and fully aligned with the expected answer. MZS was ranked last because it combines numerical calculation with critique of metric suitability in an imbalanced classification setting, making it analytically demanding despite being labelled as knowledge.

Figure 18: **Illustrative contrast SBQ sample, easy difficulty tier (max score gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SBQ Sample — Medium Difficulty (Max Score Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 8 — Planning & CSPs (Lecture Slides) | Model: Gemini-2.5-Flash | Task type: Scenario-Based Question (SBQ) | Difficulty target: Medium

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS+B)	Describe how Forward Checking would update the domains of variables B and C immediately after A is assigned “Red”.	Analyze	Understanding [†]	6	0.647
B (ZS+B)	Given that DevA is assigned to Task1, and knowing the “all different” constraint applies to both developers and tasks, apply the Minimum Remaining Values (MRV) heuristic to determine which unassigned developer should be selected next. Show the updated legal domains and explain your choice.	Application	Application	6	0.903
C (MZS)	Using the Most Constrained Variable (MRV) heuristic, which variable should the agent select next for assignment, and why?	Application	Application	4	0.903
D (OURS)	Compute the total number of initial, unconstrained colour assignments possible for all 7 regions of the map using the given colour domain.	Application	Application	7	0.965
E (ZS)	Explain two specific heuristics, one for variable selection and one for value ordering, that the company could implement to significantly improve the efficiency of their backtracking search.	Application	Application	4	0.965

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Identifies the total number of regions (variables) on the map as 7.	1
Identifies the size of the colour domain (values) as 3 (red, green, blue).	1
Applies the combinatorial principle for independent assignments, using the number of colours as the base and the number of regions as the exponent (3^7).	2
Calculates the result of 3^7 correctly.	2
States the final computed total number of initial unconstrained assignments as 2187.	1
<i>Model answer:</i> With 7 regions on the map and a colour domain size of 3 (red, green, blue), the total number of initial unconstrained assignments is calculated as 3^7 , which equals 2187.	7

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
OURS (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
MZS+B (rank 5):									
1.0	1.0	1.0	0.5	1.0	0.0	1.0	0.5	0.5	0.6

Selection Justification

Automated selection (max score gap axis): task selected from medium difficulty pool to maximise the spread between the highest- and lowest-scoring pipelines on the discrimination-weighted criterion score σ_w . **Best** σ_w : 0.9648 **Worst** σ_w : 0.6469 **Score gap**: 0.3179 **Gap percentile rank**: 96.7th **Runner-up gap**: 0.2484 **Mean** σ_w : 0.8767

Blind human evaluation ranking: OURS > ZS > ZS+B > MZS > MZS+B. OURS was ranked first because it is the most internally consistent and objectively gradable SBQ in the set: the task is a sharply defined application of combinatorial reasoning, the declared and true Bloom levels match, and the answer maps directly to a logically sequenced mark scheme. MZS+B was ranked last because it overstates the cognitive level: describing Forward Checking domain updates after a fixed assignment is procedural understanding rather than genuine analysis.

Figure 19: **Illustrative contrast SBQ sample, medium difficulty tier (max score gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†] Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Illustrative Contrast SBQ Sample — Hard Difficulty (Max Score Gap Axis)

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 9 — MDPs / Uncertainty (Slides) | Model: GPT-4.1 | Task type: Scenario-Based Question (SBQ) | Difficulty target: Hard

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS+B)	Design a suitable reward function for the robot's navigation MDP. Briefly justify your choices for how rewards should be assigned to regular, goal, and hazardous states.	Synthesis	Synthesis	5	0.702
B (ZS)	Apply the Bellman equation to compute the updated utility $U((2, 1))$ for moving left, showing your calculation, and determine which action yields the maximum expected utility.	Evaluation	Application [†]	6	0.910
C (OURS)	Assess the effectiveness of modeling the warehouse inventory management system as a Markov Decision Process, given the described state space, actions, and transition uncertainties.	Evaluation	Evaluation	8	0.965
D (MZS)	Explain the process for updating the utility value of state (2, 1) using the Bellman equation and discuss what the value 0.655 represents in this context.	Evaluation	Understanding [†]	8	0.723
E (MZS+B)	Discuss how the reward function influences the policy derived from value iteration in such an uncertain inventory system. What might be the consequences of missing or poorly designed reward components?	Evaluation	Analyze [†]	8	0.938

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
MDP formalism allows systematic modeling of sequential decisions, capturing inventory dynamics through defined states and actions.	1
Probabilistic state transitions in the MDP match real-world uncertainties such as customer demand variability and delivery timing.	2
The MDP reward structure effectively incorporates holding costs, ordering costs, and expected profits, supporting quantitative evaluation of policies.	1
The large state space may increase computational complexity, potentially limiting tractability for optimal policy computation.	2
MDP modeling is suitable for inventory management when transition probabilities and rewards are well-defined, but performance hinges on accurate probability estimation and scalable solution methods.	2
<i>Model answer:</i> MDP formalism allows systematic modeling of sequential decisions, capturing inventory dynamics through defined states and actions. Probabilistic state transitions in the MDP match real-world uncertainties such as customer demand variability and delivery timing. The MDP reward structure effectively incorporates holding costs, ordering costs, and expected profits, supporting quantitative evaluation of policies. The large state space may increase computational complexity, potentially limiting tractability for optimal policy computation. MDP modeling is suitable for inventory management when transition probabilities and rewards are well-defined, but performance hinges on accurate probability estimation and scalable solution methods.	/8

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
<i>OURS (rank 1):</i>									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
<i>MZS (rank 5):</i>									
1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.5	1.0	0.4

Selection Justification

Automated selection (max score gap axis): task selected from hard difficulty pool to maximise the spread between the highest- and lowest-scoring pipelines on the discrimination-weighted criterion score σ_w . **Best** σ_w : 0.9648 **Worst** σ_w : 0.7015 **Score gap**: 0.2633 **Gap percentile rank**: 93.8th **Runner-up gap**: 0.2565 **Mean** σ_w : 0.8474

Blind human evaluation ranking: OURS > MZS+B > ZS > ZS+B > MZS. OURS was ranked first because it is the clearest example of a genuine evaluation-level SBQ in the set: the scenario is essential, the question requires a justified judgment about MDP suitability, and the mark scheme balances strengths and limitations of the formulation. MZS was ranked last because it substantially inflates Bloom level: explaining a Bellman update and interpreting the resulting value is a conceptual understanding task, not an evaluation task.

Figure 20: **Illustrative contrast SBQ sample, hard difficulty tier (max score gap axis).** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Representative SAQ Sample — Easy Difficulty

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 7 — Heuristic Search & Adversarial Search (Slides) | Model: GPT-4.1
 | Task type: Short Answer Question (SAQ) | Difficulty target: Easy

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS+B)	Explain the role and structure of game trees in modeling two-player adversarial games, including the significance of MAX, MIN, and terminal nodes.	Understanding	Understanding	9	0.674
B (ZS+B)	Explain the structure and purpose of a game tree in the context of two-player adversarial games, naming each type of node and describing their relationships.	Understanding	Understanding	7	0.674
C (OURS)	Define what a game tree is in the context of two-player games.	Knowledge	Knowledge	3	1.000
D (ZS)	Explain the structure of a game tree in two-player adversarial games, describing the roles of MAX, MIN, and terminal nodes.	Knowledge	Understanding [†]	5	0.961
E (MZS)	Explain the roles of MAX and MIN players in a two-player adversarial game and how game trees are used to represent the sequence of possible moves.	Knowledge	Understanding [†]	7	0.634

[†]Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
A game tree is a tree structure where nodes represent game states and edges represent moves.	1
It models two-player games with alternating MAX and MIN player turns.	1
Terminal nodes represent final outcomes of the game and have no children.	1
<i>Model answer:</i> A game tree is a tree structure where nodes represent game states and edges represent moves, it models two-player games with alternating MAX and MIN player turns, and terminal nodes represent final outcomes of the game and have no children.	/3

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (rank 1):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MZS (rank 5):							
1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.8

Selection Justification

Representative matched-task selection: task selected from the easy difficulty pool ($n=7$ candidates; selected percentile = 42.9) as a representative matched-task sample rather than an extreme contrast case. **Mean σ_w :** 0.7884

Blind human evaluation ranking: OURS > ZS+B > MZS+B > ZS > MZS. OURS was ranked first because it is the most precise, controlled, and internally consistent assessment triple: the question is tightly scoped, the declared and true Bloom levels match exactly, and the mark scheme and expected answer align perfectly around a single knowledge-level objective. MZS was ranked last because, although grounded and answerable, it is less tightly structured and Bloom-misaligned: it asks for conceptual explanation of MAX/MIN roles and game-tree representation while being labelled only as knowledge.

Figure 21: **Representative SAQ sample, easy difficulty tier.** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Representative SAQ Sample — Medium Difficulty

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 7 — Heuristic Search & Adversarial Search (Answers) | Model: GPT-4.1 | Task type: Short Answer Question (SAQ) | Difficulty target: Medium

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (OURS)	Apply the concept of admissible heuristics to select an appropriate heuristic for an A* search in a new maze, ensuring the optimal path is found.	Application	Application	5	1.000
B (MZS+B)	Compare the effects of scaling an admissible heuristic by a factor greater than 1 on Greedy Best-First Search versus A* search. Why do these effects differ?	Analyze	Analyze	10	0.782
C (ZS)	Describe what is meant by an admissible heuristic and discuss how admissibility and informativeness affect the performance and guarantees of A* search.	Application	Analyze [†]	8	0.797
D (MZS)	Explain the differences in the paths chosen by Greedy Best-First Search and A* Search in the given example, including the reasons behind their respective path costs.	Application	Analyze [†]	9	0.797
E (ZS+B)	Explain the difference between Greedy Best-First Search (GBFS) and A* Search in terms of the paths and path costs they find in the given example.	Understanding	Understanding	8	0.674

[†]Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Choose a heuristic that never over-estimates the true cost from any node to the goal, satisfying admissibility.	1
Verify the selected heuristic is admissible so A* will guarantee finding the optimal path through the maze.	2
Avoid using scaled heuristics (such as multiplying by a constant greater than 1) since this may cause over-estimation and inadmissibility.	1
Consider using a commonly admissible heuristic like Manhattan distance for maze environments, as it estimates the minimum number of steps to the goal without overestimating.	1
<i>Model answer:</i> Select a heuristic that never over-estimates the true cost from any node to the goal to ensure admissibility, verify this property to guarantee that A* finds the optimal path, do not use scaled heuristics with constants greater than 1 as they may over-estimate, and use an admissible heuristic like Manhattan distance for a maze since it does not overestimate the number of steps to the goal.	/5

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
OURS (rank 1):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ZS (rank 5):							
1.0	1.0	1.0	1.0	1.0	0.5	1.0	0.8

Selection Justification

Representative matched-task selection: task selected from the medium difficulty pool ($n=35$ candidates; selected percentile = 48.6) as a representative matched-task sample rather than an extreme contrast case. **Mean σ_w :** 0.8100

Blind human evaluation ranking: OURS > MZS+B > ZS+B > MZS > ZS. OURS was ranked first because it is the most complete and internally consistent assessment triple: it poses a genuine application-level task, operationalizes admissibility concretely, and supports highly objective grading through a tightly matched mark scheme and answer. ZS was ranked last because it is Bloom-misaligned and too abstract for its declared task type: it frames a broad conceptual discussion of admissibility and informativeness rather than a concrete application to a specific case.

Figure 22: **Representative SAQ sample, medium difficulty tier.** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Representative SAQ Sample — Hard Difficulty

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2 — Introduction to Machine Learning & Unsupervised Learning (Lecture Notes with Solutions) | Model: GPT-4o-mini | Task type: Short Answer Question (SAQ) | Difficulty target: Hard

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS)	Compare and contrast the Single-Linkage, Complete-Linkage, and Average-Linkage criteria in hierarchical clustering, focusing on how they define the distance between clusters.	Evaluation	Analyze [†]	13	0.634
B (MZS+B)	Given a practical scenario where delivery by helicopter is limited to a 100-mile range, justify why Complete-Linkage clustering would be preferable over other linkage criteria for forming suitable clusters.	Evaluation	Evaluation	10	1.000
C (ZS+B)	Compare and contrast the single-linkage, complete-linkage, and average-linkage criteria used in hierarchical clustering to measure cluster distances.	Analyze	Analyze	10	1.000
D (OURS)	Assess the effectiveness of running K-means multiple times with different initialisations or using k-means++ to avoid local minima, as described in the text.	Evaluation	Evaluation	8	1.000
E (MZS)	Given a practical scenario where a helicopter delivery service must cluster cities such that any city in a cluster is reachable within a 100-mile flight, discuss why the Complete-Linkage criterion is the suitable choice and how it ensures this constraint.	Evaluation	Application [†]	10	0.595

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Mark scheme and expected answer (MZS+B — rank 1):

Marking point	Marks
Complete-Linkage clustering forms clusters based on the maximum distance between points, ensuring all points in a cluster are within a certain range.	2
This method tends to produce compact, tight clusters, which aligns with the helicopter’s 100-mile delivery range constraint.	2
In contrast, Single-Linkage clustering can create elongated clusters where some points are far apart, possibly exceeding the 100-mile limit.	1
Average-Linkage clustering averages distances, which may allow some points in a cluster to be beyond the 100-mile range even if the average distance is acceptable.	1
Complete-Linkage minimizes the risk of any delivery exceeding the 100-mile range, which is critical for operational feasibility and safety.	2
Using Complete-Linkage supports clear decision-making by providing well-defined cluster boundaries that respect the practical distance limit.	1
Overall, Complete-Linkage clustering better guarantees deliveries stay within the helicopter’s operational range compared to other linkage methods.	1

Model answer: Complete-Linkage clustering forms clusters based on the maximum distance between points, ensuring all points within a cluster remain within the helicopter’s 100-mile delivery range. This method produces compact, tight clusters, aligning closely with the operational constraint, unlike Single-Linkage clustering which can generate elongated clusters with points exceeding the range. Average-Linkage, by averaging distances, may also include points beyond the limit despite an acceptable average. Therefore, Complete-Linkage minimizes the risk of any delivery exceeding the range, supports clear decision-making through well-defined cluster boundaries, and overall better guarantees feasible and safe helicopter deliveries. **/10**

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	BLA
MZS+B (rank 1):							
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MZS (rank 5):							
1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.6

Selection Justification

Representative matched-task selection: task selected from the hard difficulty pool ($n=17$ candidates; selected percentile = 47.1) as a representative matched-task sample rather than an extreme contrast case. **Mean σ_w :** 0.8458

Blind human evaluation ranking: MZS+B > OURS > ZS+B > ZS > MZS. MZS+B was ranked first because it is a clear, well-grounded evaluation task embedded in a realistic constraint-driven scenario; it requires comparison of linkage criteria, judgment under the 100-mile range constraint, and an explicitly justified decision. MZS was ranked last because it underdelivers on evaluation: although grounded in a plausible scenario, it is closer to application than genuine evaluative reasoning and does not sufficiently require comparison across methods under competing considerations.

Figure 23: **Representative SAQ sample, hard difficulty tier.** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†] Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Representative SBQ Sample — Easy Difficulty

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2–3 — Introduction to Machine Learning & Unsupervised Learning (Annotated Notes) | Model: GPT-4.1 | Task type: Scenario-Based Question (SBQ) | Difficulty target: Easy

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (OURS)	Identify the two data splits used for model development in the scenario.	Knowledge	Knowledge	2	0.965
B (ZS+B)	Define the metrics MAE, MSE, and RMSE in the context of regression, and explain how each metric measures prediction errors.	Understanding	Understanding	5	0.965
C (MZS+B)	Describe the roles of the training set and the test set in supervised learning.	Knowledge	Knowledge	3	0.965
D (ZS)	Which terms describe these two parts of the dataset, and why is it important to keep the evaluation set labels hidden during training?	Knowledge	Analyze [†]	4	0.685
E (MZS)	If the algorithm always predicts “no disease,” calculate its accuracy, and discuss whether this is a good way to judge the algorithm’s performance in this scenario.	Knowledge	Analyze [†]	5	0.685

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Scenario (ZS+B — rank 1):

A data scientist builds a regression model to predict housing prices based on several features. After training the model, she evaluates it on a test set and calculates the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root-Mean-Squared Error (RMSE).

Mark scheme and expected answer (ZS+B — rank 1):

Marking point	Marks
Defines Mean Absolute Error (MAE) as the average absolute difference between predicted and true values.	1
Defines Mean Squared Error (MSE) as the average squared difference between predicted and true values.	1
Defines Root-Mean-Squared Error (RMSE) as the square root of the average squared difference.	1
Explains that MAE measures average magnitude of errors, MSE penalizes larger errors, and RMSE brings scale back to original units.	2
<i>Model answer:</i> Mean Absolute Error (MAE) is calculated as the average absolute difference between the predicted and actual housing prices. Mean Squared Error (MSE) measures the average of the squared differences, thus penalizing larger errors more heavily. Root-Mean-Squared Error (RMSE) is the square root of the MSE, providing an error value in the same units as the housing prices. MAE reflects the average error magnitude, MSE highlights the impact of large deviations, and RMSE is useful for interpreting error size in original terms.	/5

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
ZS+B (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
MZS (rank 5):									
1.0	1.0	0.5	1.0	1.0	0.0	1.0	0.5	1.0	0.4

Selection Justification

Representative matched-task selection: task selected from the easy difficulty pool ($n=11$ candidates; selected percentile = 45.5) as a representative matched-task sample rather than an extreme contrast case. **Mean σ_w :** 0.8527

Blind human evaluation ranking: ZS+B > MZS+B > OURS > ZS > MZS. ZS+B was ranked first because it strikes the best balance between scenario use, conceptual value, and internal consistency: it poses a clean understanding-level task, uses a realistic regression scenario proportionately, and provides a clear, complete, and highly objective mark scheme. MZS was ranked last because it is Bloom-misaligned and cognitively broader than its label suggests: it combines numerical calculation with critique of metric suitability in an imbalanced disease-detection setting while being declared only as knowledge.

Figure 24: **Representative SBQ sample, easy difficulty tier.** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Representative SBQ Sample — Medium Difficulty

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 2–3 — Introduction to Machine Learning & Unsupervised Learning (Annotated Notes) | Model: GPT-4o-mini | Task type: Scenario-Based Question (SBQ) | Difficulty target: Medium

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (ZS)	Explain why a model that always predicts “no disease” can achieve 99% accuracy on this problem but is not useful. What alternative metrics should be considered to properly evaluate the model’s effectiveness?	Application	Understanding [†]	8	0.813
B (OURS)	Differentiate between the evaluation approaches used for the classification task and the regression task described in the scenario.	Analyze	Analyze	7	1.000
C (MZS+B)	Compare the implications of using MAE versus MSE as evaluation metrics for this regression model, particularly in the presence of outliers in the data.	Analyze	Analyze	7	0.861
D (MZS)	Explain why the high accuracy achieved by this model might be misleading in evaluating its performance. Which alternative metrics would provide better insight into the model’s effectiveness at detecting disease D, and why?	Application	Analyze [†]	8	0.973
E (ZS+B)	Analyse which metric, precision or sensitivity, would highlight the weakness of the algorithm that always predicts “no disease”, and justify why.	Analyze	Analyze	4	0.840

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Scenario (OURS — rank 1):

A supervised learning algorithm is developed to predict the presence of a rare disease D using chest X-ray images. The data is split into a training set, which includes labeled examples, and a test set, where labels are hidden during evaluation. The disease occurs in 1% of the population represented in the dataset. Upon evaluation, an accuracy of 99% was recorded on the test set. The test set maintains the same class distribution as the population, where the positive class is rare. Additionally, alternative classification metrics such as precision, sensitivity, and F1 score are considered for further analysis. In a separate regression task, predictions consist of real-valued outputs compared against true target values using error metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root-Mean-Squared Error (RMSE).

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Classification uses discrete class labels with metrics like accuracy, precision, sensitivity, and F1 score to evaluate prediction quality.	2
Accuracy can be misleading in classification when classes are imbalanced, such as a rare disease with 1% prevalence.	1
Regression evaluates continuous outputs using error metrics like MAE, MSE, and RMSE to measure prediction errors.	2
Classification metrics assess categorical correctness and balance of false positives/negatives, while regression metrics quantify numerical deviations.	2

Model answer: Classification uses discrete class labels with metrics like accuracy, precision, sensitivity, and F1 score to evaluate prediction quality. Accuracy can be misleading in classification when classes are imbalanced, such as a rare disease with 1% prevalence. Regression evaluates continuous outputs using error metrics like MAE, MSE, and RMSE to measure prediction errors. Classification metrics assess categorical correctness and balance of false positives and negatives, while regression metrics quantify numerical deviations. //

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
OURS (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ZS (rank 5):									
1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.8

Selection Justification

Representative matched-task selection: task selected from the medium difficulty pool ($n=30$ candidates; selected percentile = 50.0) as a representative matched-task sample rather than an extreme contrast case. **Mean σ_w :** 0.8974

Blind human evaluation ranking: OURS > ZS+B > MZS > MZS+B > ZS. OURS was ranked first because it is the most internally consistent, scenario-grounded, and cognitively well-targeted item in the set: it asks for a genuine analyze-level comparison, makes full use of both the classification and regression parts of the scenario, and supports highly consistent grading. ZS was ranked last because it overstates the cognitive level: despite a useful scenario, the question mainly asks for conceptual explanation of why 99% accuracy is misleading and which alternative metrics matter, making it closer to understanding than true application.

Figure 25: **Representative SBQ sample, medium difficulty tier.** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies the true cognitive demand, as identified by the blind human evaluator.

Representative SBQ Sample — Hard Difficulty

Course: 4CCSAIAI Introduction to Artificial Intelligence | Source: Week 4 — Ethics (Slides) | Model: Gemini-2.5-Flash | Task type: Scenario-Based Question (SBQ) | Difficulty target: Hard

Pipeline	Question stem	Declared Bloom	True Bloom	Marks	σ_w
A (MZS+B)	Identify and explain at least two distinct causes of bias (from “World bias”, “Representation bias”, “Measurement bias”, “Algorithm bias”, “Evaluation bias”) that likely contributed to the problematic auto-crop behavior observed in Rui’s system.	Analyze	Analyze	6	0.965
B (ZS)	Identify and explain two significant ethical harms that the Epimetheus app could cause, and propose two distinct mitigation strategies the developers could implement to address these harms, particularly concerning personal identity.	Evaluation	Analyze [†]	8	0.938
C (OURS)	Develop a structured set of guidelines for future AI feature development at the social media company, explicitly addressing the ethical concerns (stereotypes, indirect discrimination, objectification) identified in the auto-crop feature’s review.	Synthesis	Synthesis	12	0.965
D (ZS+B)	From an AI ethics perspective, explain two potential societal harms that “Sentinel-Eye” could inflict, even if it successfully reduces overt crime.	Analyze	Understanding [†]	4	0.849
E (MZS)	What specific measures can the developers of Epimetheus implement to address the issue of gender misidentification and its associated dignitary harms, particularly for transgender individuals, as discussed in the course material?	Evaluation	Application [†]	8	0.910

[†] Declared–true Bloom mismatch identified by blind human evaluator.

Scenario (OURS — rank 1):

A software engineer at a social media company developed an ‘auto-crop’ feature that automatically cropped images attached to posts. The system was configured to identify and focus on specific image areas. A deep neural network was trained on a dataset containing images and their associated saliency data. Following its deployment, the feature led to an increase in posts containing images and an uplift in user engagement metrics, including time spent on the platform and the number of clicks. However, subsequent observations by a group of users showed systematic patterns within the auto-crop’s behavior. These patterns included a consistent focus on White individuals over Black individuals, famous men over famous women, and a frequent centering on women’s chest areas. An internal ethics review stated that this behavior exhibited characteristics concerning stereotypes, indirect discrimination, and the objectification of women.

Mark scheme and expected answer (OURS — rank 1):

Marking point	Marks
Mandate a rigorous, pre-deployment audit process for all AI training datasets, ensuring they are demographically representative and balanced to actively prevent the emergence of stereotypes and indirect discrimination.	3
Implement ethically-guided design principles for attention or saliency mechanisms in image processing AI, specifically to avoid hyper-focus on sensitive body areas and prevent objectification, perhaps using constrained optimization or negative exemplars.	3
Integrate a comprehensive bias detection and mitigation framework into the AI development lifecycle, applying quantitative fairness metrics (e.g., demographic parity, equalized odds) at multiple stages from data preparation to model deployment.	2
Establish a standing, interdisciplinary ethical AI review board, comprising engineers, ethicists, and diverse user representatives, responsible for assessing the societal impact and potential biases of new AI features from conception.	2
Develop transparent channels for user feedback specifically on AI behavior, coupled with a swift, accountable system for investigating, communicating, and rectifying identified ethical violations in deployed features.	2
<i>Model answer:</i> Future AI feature development will mandate a rigorous, pre-deployment audit process for all AI training datasets, ensuring they are demographically representative and balanced to actively prevent the emergence of stereotypes and indirect discrimination. Ethically-guided design principles will be implemented for attention or saliency mechanisms in image processing AI, specifically to avoid hyper-focus on sensitive body areas and prevent objectification, perhaps using constrained optimization or negative exemplars. A comprehensive bias detection and mitigation framework will be integrated into the AI development lifecycle, applying quantitative fairness metrics like demographic parity or equalized odds at multiple stages from data preparation to model deployment. Furthermore, a standing, interdisciplinary ethical AI review board, comprising engineers, ethicists, and diverse user representatives, will be established, responsible for assessing the societal impact and potential biases of new AI features from conception. Finally, transparent channels for user feedback specifically on AI behavior will be developed, coupled with a swift, accountable system for investigating, communicating, and rectifying identified ethical violations in deployed features.	/12

Per-criterion scores (best and worst pipeline):

Grounding	Clarity	Integrity	Gradability _{stem}	Gradability _{triple}	Mk. scheme	Fidelity	SRN	SG	BLA
OURS (rank 1):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0
ZS+B (rank 5):									
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.5	0.6

Selection Justification

Representative matched-task selection: task selected from the hard difficulty pool ($n=16$ candidates; selected percentile = 50.0) as a representative matched-task sample rather than an extreme contrast case. **Mean** σ_w : 0.9252

Blind human evaluation ranking: **OURS** > **MZS+B** > **ZS** > **MZS** > **ZS+B**. **OURS** was ranked first because it best combines scenario use, cognitive demand, and assessment structure: the auto-crop case is essential rather than decorative, the task requires genuine synthesis, and the mark scheme is both concrete and flexible enough to support higher-order ethical reasoning. **ZS+B** was ranked last because, although clear and grounded, it is more explanatory than analytical: asking for two harms of Sentinel-Eye is better characterized as understanding than true analysis, making it less discriminative as a representative hard SBQ.

Figure 26: **Representative SBQ sample, hard difficulty tier.** Pipeline identities are shown explicitly: ZS = zero_shot, ZS+B = zero_shot_bloom, MZS = multi_stage_zero_shot, MZS+B = multi_stage_zero_shot_bloom, and Ours = proposed pipeline. σ_w denotes the discrimination-weighted composite criterion score (higher is better, max = 1). The mark scheme and per-criterion breakdown are shown for the rank-1 and rank-5 pipelines only. [†]Flags items where the declared Bloom level under-specifies or overstates the true cognitive demand, as identified by the blind human evaluator.

Evaluation Type	QType	N	Prop.	Base.	Rate
Representative	SAQ	3	2	1	66.7%
Representative	SBQ	3	2	1	66.7%
Illus. Contrast (msg)	SAQ	3	2	1	66.7%
Illus. Contrast (msg)	SBQ	3	2	1	66.7%
Illus. Contrast (bbg)	SAQ	3	3	0	100%
Illus. Contrast (bbg)	SBQ	3	2	1	66.7%
Overall	All	18	13	5	72.2%

Table 14: Aggregate qualitative selection outcomes across all 18 matched-task manual comparisons. msg = max_score_gap; bbg = bloom_bla_gap. “Prop.” and “Base.” count the number of settings in which the proposed pipeline or any baseline ranked first respectively.

Difficulty	1st	2nd	3rd	4th	5th
Easy	Ours	ZS+B	MZS+B	ZS	MZS
Medium	Ours	MZS+B	ZS+B	MZS	ZS
Hard	MZS+B	Ours	ZS+B	ZS	MZS

Table 15: Pipeline rankings for representative SAQ evaluations by difficulty level.

Difficulty	1st	2nd	3rd	4th	5th
Easy	ZS+B	MZS+B	Ours	ZS	MZS
Medium	Ours	ZS+B	MZS	MZS+B	ZS
Hard	Ours	MZS+B	ZS	MZS	ZS+B

Table 16: Pipeline rankings for representative SBQ evaluations by difficulty level.

Difficulty	1st	2nd	3rd	4th	5th
Easy	Ours	ZS+B	MZS+B	ZS	MZS
Medium	ZS+B	Ours	MZS+B	MZS	ZS
Hard	Ours	MZS+B	ZS	MZS	ZS+B

Table 17: Pipeline rankings for SAQ illustrative contrasts on the max_score_gap axis.

Difficulty	1st	2nd	3rd	4th	5th
Easy	Ours	ZS	MZS+B	ZS+B	MZS
Medium	Ours	ZS+B	MZS	ZS	MZS+B
Hard	Ours	MZS+B	ZS	MZS	ZS+B

Table 18: Pipeline rankings for SAQ illustrative contrasts on the bloom_bla_gap axis.

Difficulty	1st	2nd	3rd	4th	5th
Easy	MZS+B	ZS+B	Ours	ZS	MZS
Medium	Ours	ZS	ZS+B	MZS	MZS+B
Hard	Ours	MZS+B	ZS	ZS+B	MZS

Table 19: Pipeline rankings for SBQ illustrative contrasts on the max_score_gap axis.

Difficulty	1st	2nd	3rd	4th	5th
Easy	MZS+B	ZS+B	Ours	ZS	MZS
Medium	Ours	ZS+B	ZS	MZS	MZS+B
Hard	Ours	ZS+B	MZS+B	ZS	MZS

Table 20: Pipeline rankings for SBQ illustrative contrasts on the bloom_bla_gap axis.

Intent vs. Surface: Recovering Acoustic Realization from Modern ASR for Pronunciation Training

Seongjin Park

Speak

360 Spear St. 4F. San Francisco, California. 94105.

seongjin@usespeak.com

Abstract

Pronunciation feedback in language learning depends on accurate detection of learner errors, but it is unclear whether modern ASR systems are suitable for this purpose. Their language models recover intended words rather than what was actually pronounced, systematically masking mispronunciations. This is a tendency we call *intent bias*. By evaluating eight ASR systems spanning three architectures on two L2 English corpora, we find that overcorrection rate correlates inversely with word error rate. In other words, ASR systems with lower WER tend to mask more pronunciation errors. We propose surface-faithful reranking, an inference-time method that uses phoneme-level acoustic similarity to select N-best hypotheses closer to what the learner actually said. Without retraining or access to model internals, the method reduces the false acceptance rate of mispronunciations by 6.0 percentage points on L2-ARCTIC and 5.6 on speechocean762. The improvement is consistent across age groups and first-language backgrounds, though substantial overcorrection remains, pointing to the need for pronunciation-aware ASR objectives.

1 Introduction

Pronunciation is among the most difficult aspects of second-language acquisition, and automated feedback has long been sought as a scalable complement to human tutors (Eskenazi, 2009; Neri et al., 2002). At the core of most computer-assisted pronunciation training (CAPT) systems lies an automatic speech recognition component whose transcriptions are compared against expected text to identify mispronounced words (Witt and Young, 2000). The accuracy of this detection step is therefore critical for providing useful feedback to learners.

Modern ASR systems, however, are optimized to report what the speaker most likely *meant* rather

than what was actually produced, since their language model components assign high probability to grammatically and lexically plausible sequences. This results in effectively correcting non-standard pronunciations before they reach the downstream tasks. For example, when a learner says “she coming” (omitting “is”), the ASR often produces “she is coming”. Also, when a learner produces something closer to “sheep” while reading “ship,” the system may output “ship” based on the context. We refer to this tendency as *intent bias* throughout the paper. In our analysis of the speechocean762 (Zhang et al., 2021) corpus using Parakeet, 82.0% of utterances contain at least one overcorrected mispronunciation in the top-ranked hypothesis, and the system tends to be most confident when it overcorrects.

Thus, we propose *surface-faithful reranking*, a method that works entirely at inference time to recover what the learner actually pronounced. Given the N-best hypotheses from any ASR system, we obtain a reference phoneme sequence from an independent phoneme recognizer, then select the hypothesis whose phonetic transcription is closest to that reference. This reranked hypothesis drives mispronunciation detection while the original top-1 output is retained for display, allowing a single system to serve both readability and diagnostic accuracy. On L2-ARCTIC (Zhao et al., 2018), the method reduces the false acceptance rate on severe mispronunciations from 80.8% to 74.8%, and on speechocean762 test, from 70.6% to 65.0%. The improvements are consistent across age groups (speechocean762) and across all six L1 backgrounds tested (L2-ARCTIC).

The contributions of this work are threefold. First, we show that overcorrection is not incidental but systematic: across eight ASR systems spanning three architectures (autoregressive, hybrid CTC-RNNT, CTC-only), lower word error rate correlates with higher overcorrection, suggesting that

WER alone is insufficient for evaluating ASR in educational contexts. Second, we propose surface-faithful reranking, an inference-time method that partially mitigates this bias without retraining or white-box access. Third, we evaluate across two corpora, multiple age groups, and diverse L1 backgrounds, showing consistent gains.

2 Related Work

2.1 Mispronunciation Detection in CAPT

Goodness of Pronunciation (GOP), introduced by Witt and Young (2000), remains foundational in CAPT. GOP computes the posterior probability ratio between forced-aligned and freely decoded phone sequences, providing a frame-level pronunciation score. It requires white-box access to the acoustic model and frame-level forced alignment, assumptions that hold for HMM-GMM or hybrid systems but not for modern end-to-end ASR or commercial APIs.

End-to-end approaches train neural networks directly on L2 speech for mispronunciation detection and diagnosis (MDD) (Leung et al., 2019; Li et al., 2017). More recently, self-supervised representations from wav2vec2 (Baevski et al., 2020) have been fine-tuned for this task as well (Xu et al., 2021; Peng et al., 2021), achieving competitive results with less labeled data. Korzekwa et al. (2021) incorporate uncertainty modeling to improve detection robustness. However, all these approaches require annotated non-native speech corpora for training, data that remains scarce for most language pairs, and produce task-specific models that must be retrained for new domains.

Transformer-based assessment systems such as GOPT (Gong et al., 2022) predict pronunciation scores at multiple granularities, but their training on specific corpora limits zero-shot generalization. The method we propose avoids these constraints, as it requires no training on L2 data, no access to ASR internals, and no task-specific model.

2.2 ASR for L2 Speech and Language Model Bias

ASR systems trained on native speech perform worse on accented and non-native speech (Zechner et al., 2009). This degradation is well documented, but a subtler problem has received less attention. When ASR *succeeds* on L2 speech, it often does so by correcting the learner’s errors rather than faithfully transcribing them. This overcorrection is a

natural consequence of optimizing the model for low word error rate, which incentivizes recovering canonical word sequences regardless of acoustic evidence to the contrary.

Cámara-Arenas et al. (2023) formalize this as a conflict between two necessary conditions for pronunciation assessment. ASR must recognize diverse speech (requiring robustness) but must also refrain from normalizing non-standard pronunciations (requiring sensitivity). They show that Google ASR cannot satisfy both conditions simultaneously. Won (2025) provides further empirical evidence that WER correlates poorly with human pronunciation judgments for high-performing ASR systems. Chen et al. (2024) demonstrate that acoustic information is often overridden by language model priors, and propose training-time modifications. Our work shares the diagnosis but differs in approach. Rather than modifying training, we intervene at inference time and provide quantitative evidence of the WER-overcorrection trade-off across architectures.

2.3 N-best Reranking

N-best list reranking has been studied extensively for improving ASR accuracy, typically using language models or neural rescorers to select the most linguistically plausible hypothesis (Salazar et al., 2020). In the present work, we try to invert this paradigm. Instead of reinforcing linguistic plausibility, we use phonetic signals to select hypotheses that best match actual pronunciation. We hypothesize that surface-faithful candidates often already exist in the N-best list but are suppressed by the language model’s preference for canonical forms.

3 Method

3.1 Problem Formulation

Standard ASR systems tend to choose the hypothesis that maximizes a combination of acoustic likelihood and language model probability. For CAPT, however, this objective is often misaligned. A hypothesis that restores a dropped word or corrects a vowel substitution scores well by conventional metrics but obscures the pronunciation error a learner needs to know about. In this study, we seek an alternative selection criterion that favors hypotheses reflecting what was acoustically produced (i.e., the *surface realization*) rather than what was linguistically intended.

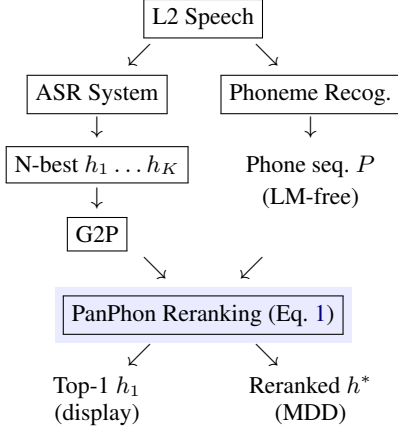


Figure 1: Surface-faithful reranking pipeline. PanPhon distance selects the hypothesis closest to the phoneme recognizer output for MDD; the top-1 hypothesis is retained for display.

3.2 Surface-Faithful Reranking

Given an utterance, we extract two independent signals, which are (1) an N-best list of text hypotheses $\mathcal{H} = \{h_1, \dots, h_K\}$ from the ASR system, and (2) a phoneme sequence P from a separate phoneme recognizer that operates without a language model. The phoneme recognizer provides a language-model-free estimate of the surface realization.

Each hypothesis h is converted to an IPA phoneme sequence using grapheme-to-phoneme (G2P) system via g2p-en (Park and Kim, 2019). We then select the hypothesis whose phonetic representation most closely matches the recognized phoneme sequence:

$$h^* = \arg \min_{h \in \mathcal{H}} d_{\text{PanPhon}}(\text{G2P}(h), P) \quad (1)$$

where d_{PanPhon} is the weighted articulatory feature edit distance described below. The selected hypothesis h^* serves as the surface-faithful transcription for mispronunciation detection, while the original top-1 hypothesis h_1 is retained for display to the learner. This dual-output design allows a single ASR system to provide both a readable transcript and accurate pronunciation diagnostics. Figure 1 illustrates the complete pipeline.

3.3 Phonetic Distance via PanPhon

Standard edit distance treats all phoneme substitutions equally, but pronunciation errors often involve phonetically similar sounds. A learner who

produces /t/ instead of /θ/ (dental fricative simplification, common among many L1 backgrounds) has made a smaller error than one who substitutes /m/ for /θ/. PanPhon (Mortensen et al., 2016) captures this by representing each phoneme as a 24-dimensional binary vector encoding articulatory features (place, manner, voicing, nasality, and others). The substitution cost between two phonemes is their Hamming distance in this feature space. This representation provides implicit severity weighting: substitutions between acoustically similar phones incur lower costs than those between dissimilar phones.

All feature weights are fixed based on phonological theory and are not tuned on evaluation data. This avoids overfitting and ensures that the method generalizes across datasets without adaptation.

3.4 Word-Level Mispronunciation Detection

After selecting the surface-faithful hypothesis h^* , we align it against the canonical prompt (the text the learner was asked to read) using word-level edit distance. Any word that is substituted, deleted, or inserted relative to the prompt is flagged as a detected mispronunciation. This alignment-based approach requires no additional thresholds or classifiers.

4 Experimental Setup

4.1 Datasets

We evaluate on two L2 English corpora that differ in speaker demographics and annotation style.

L2-ARCTIC (Zhao et al., 2018) contains 26,867 read-speech utterances from 24 adult L2 English speakers across six L1 backgrounds (Arabic, Chinese, Hindi, Korean, Spanish, and Vietnamese). Each utterance has manual phoneme-level annotations marking substitutions, deletions, and insertions. We derive word-level mispronunciation labels by marking any word containing at least one phone-level error. Following the annotation guidelines, we categorize errors by severity: *severe* errors affect intelligibility (e.g., vowel substitution /t/→/i/ in “ship” heard as “sheep”, or consonant deletion making a word unrecognizable), *moderate* errors have uncertain impact on comprehension, and *acceptable* variations are accent features that do not impair intelligibility (e.g., dental fricative simplification /θ/→/t/).

Speechocean762 (SO762) (Zhang et al., 2021) contains 5,000 utterances from Mandarin L1 speakers including children (under 13), teenagers (13–17), and adults (18 and older), with word-level and phone-level accuracy scores on a 0–2 scale. We derive mispronunciation labels from phone-level scores: a word is mispronounced if any of its phones has accuracy below 2.0. For severity analysis, we distinguish *severe* errors (minimum phone accuracy < 1.0 , indicating a phone produced so incorrectly that it is not recognizable as the target) from *moderate* errors ($1.0 \leq$ minimum accuracy < 2.0 , indicating a noticeable but identifiable deviation). We report main results on the SO762 test split (samples 2500–4999, $n=2,497$ utterances) for direct comparability with GOPT (Gong et al., 2022), which was trained on the SO762 training split. The full 5,000-utterance set is used only in the WER–FAR correlation analysis of §5.2 where it is explicitly disclaimed.

4.2 ASR Systems and Baselines

Our primary ASR system is NVIDIA Parakeet, a FastConformer-based hybrid CTC-RNNT model (Majumdar and Koluguri, 2024), from which we extract $K=10$ N-best hypotheses. As a second system, we use Whisper large-v2 (Radford et al., 2023) via faster-whisper (SYSTRAN, 2023), also with $K=10$ hypotheses from beam search. Testing on two architecturally different systems (Parakeet uses hybrid CTC-RNNT decoding while Whisper uses autoregressive decoding with a stronger internal language model) lets us examine how architectural choices affect both intent bias and the potential for reranking.

The phoneme recognizer is wav2vec2-lv-60-espeak-cv-ft, a multilingual CTC-based model fine-tuned on Common Voice (Baevski et al., 2020). Its language-model-free decoding produces phoneme sequences that reflect acoustic content without linguistic correction.

Our method treats the ASR system as a black box. In other words, only the N-best text hypotheses and their ranks are used, and no internal scores, alignments, or posteriors are used for reranking.

To examine how overcorrection varies with ASR accuracy, we additionally evaluate Whisper tiny, base, small, and medium (39M to 769M parameters), wav2vec2-base-960h and wav2vec2-large-960h (Baevski et al., 2020), both CTC-only models without any language model component. This gives us eight systems spanning three architectures: au-

toregressive (Whisper family), hybrid CTC-RNNT (Parakeet), and CTC-only (wav2vec2).

We compare against: (1) ASR top-1 baselines across all systems; (2) Kaldi-based GOP (Witt and Young, 2000), which requires white-box access to acoustic model posteriors; (3) GOPT (Gong et al., 2022), a transformer trained on SO762 for pronunciation scoring; (4) word-level ASR confidence thresholding, which flags low-confidence words as mispronounced (oracle threshold); and (5) direct phoneme comparison, which bypasses ASR entirely and compares G2P of canonical text against phoneme recognizer output using PanPhon distance (oracle threshold).

4.3 Evaluation Metrics

We report two complementary metrics following Li et al. (2017). *False Acceptance Rate* (FAR) measures overcorrection, which represents the proportion of mispronounced words that the system incorrectly accepts as correct. High FAR means learner errors go undetected. *False Rejection Rate* (FRR) measures false alarms, and it represents the proportion of correctly pronounced words incorrectly flagged as errors. High FRR annoys learners by providing misleading feedback. All metrics use word-level aggregation across the full evaluation set. Statistical significance is assessed using McNemar’s test, and confidence intervals are obtained via bootstrap resampling (10,000 iterations).

5 Results

5.1 Main Results

Table 1 presents the main results across both datasets. The top-1 hypothesis exhibits high FAR on both: 80.8% on L2-ARCTIC severe and 70.6% on SO762 test. Note also the inverse trend within the SO762 column: stronger language models (Whisper large-v2, Parakeet) yield lower WER but higher FAR, foreshadowing the architecture-level correlation in §5.2.

Surface-faithful reranking reduces Parakeet’s FAR by 6.0pp on L2-ARCTIC severe and 5.6pp on SO762 test ($p < 0.001$ for both), at a modest FRR cost (1.7pp and 3.9pp). Residual FAR remains high (74.8% on L2-ARCTIC severe), indicating reranking is one component of a complete CAPT pipeline rather than a standalone solution.

To contextualize this gain, we examined paired word-level decisions on SO762 test. For every word that reranking newly accepts as correct, it

Table 1: Mispronunciation detection results (%). L2-ARCTIC: severe errors ($n_{\text{misp}}=4,183$); SO762 test: $n_{\text{utt}}=2,497$, $n_{\text{misp}}=6,066$. Top-1 WER varies inversely with FAR across architectures, consistent with intent bias (§5.2). *Kaldi GOP requires white-box AM access; no transcription, so WER not applicable. †GOPT was trained on the SO762 train split. ‡Oracle thresholds. + *Ours* = surface-faithful reranking via PanPhon distance. Dashes indicate not evaluated on that dataset.

Method	L2-ARCTIC severe			SO762 test		
	WER	FAR	FRR	WER	FAR	FRR
Kaldi GOP*	–	20.4	74.8	–	6.0	3.0
GOPT†	–	–	–	–	34.0	54.8
Confidence‡	–	56.9	28.5	–	50.4	35.0
Phoneme-only‡	–	22.7	59.6	–	15.8	67.4
wav2vec2-base-960h	22.7	63.1	14.0	49.0	41.0	32.1
wav2vec2-large-960h	18.9	67.2	10.9	45.0	44.8	28.4
Whisper tiny	21.7	66.6	12.3	35.9	55.8	20.0
Whisper large-v2	9.2	82.7	5.1	16.9	74.2	8.4
Whisper + Ours	9.8	80.9	5.3	20.0	72.4	8.2
Parakeet Top-1	8.0	80.8	5.9	19.2	70.6	8.4
Parakeet + Ours	11.8	74.8	7.6	24.9	65.0	12.3

correctly rejects 3.7 previously accepted mispronunciations (Fixed:Hurt = 3.66:1, McNemar exact OR = 3.66, 95% CI [3.01, 4.49], $p < 0.001$). Detection recall rises from 29.4% to 35.0%, a 19% relative increase that closes about 41% of the $K=10$ oracle gap ($5.6 / (70.6 - 56.9)$).

The cluster-bootstrap 95% CI for ΔFAR , resampling by utterance over 10,000 iterations, is $[-6.4, -4.9]$ on SO762 test, compared with $[-6.8, -5.0]$ on L2-ARCTIC severe. The trade-off is asymmetric: each percentage point of FAR reduction comes at roughly 0.7pp of FRR increase, concentrated in short function words (§6.6). Whisper shows a smaller 1.8pp improvement on both datasets, likely due to limited N-best diversity (§6.3).

On L2-ARCTIC, GOP achieves low FAR (20.4%) but extremely high FRR (74.8%), flagging three out of four correctly pronounced words as errors. For CAPT, persistent false alarms would erode learner trust and motivation (Neri et al., 2002). GOP also requires white-box access to acoustic model internals and frame-level forced alignment, which makes it incompatible with modern API-based ASR services. Our method offers the opposite profile within the black-box paradigm, moderate FAR with low FRR.

5.2 Overcorrection Scales with ASR Accuracy

To examine whether overcorrection is a general property of ASR optimization, we measured WER and FAR for eight systems spanning three architectures on the full SO762 set (Figure 2; num-

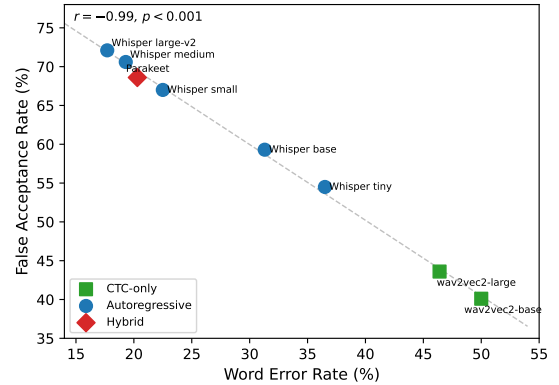


Figure 2: WER vs. overcorrection rate (FAR) on the full SO762 set ($n_{\text{misp}} \approx 12,235$) across eight ASR systems spanning three architectures. Computed on the full set for correlation power; the test-split subset yields the same direction with wider CI. Lower WER correlates strongly with higher overcorrection ($r=-0.99$, $p<0.001$).

bers in this subsection use the full set for correlation power, in contrast to the test-only Table 1). We found a very strong trend between WER and FAR ($r=-0.99$ ($p<0.001$), Spearman $\rho=-1.00$ ($p<0.001$)). Wav2vec2-base-960h, a CTC-only model with no language model, has the highest WER (50.0%) but the lowest FAR (40.1%), making many transcription errors but rarely masking mispronunciations. Whisper large-v2, with its strong autoregressive decoder, achieves the lowest WER (17.7%) but the highest FAR (72.1%). Parakeet (hybrid CTC-RNNT, WER 20.3%, FAR 68.6%) and four smaller Whisper variants (tiny, base, small, medium) fall in between.

The trend holds across all three architecture families. Even within the CTC-only pair, the larger wav2vec2 model achieves lower WER (46.4% vs. 50.0%) but higher FAR (43.6% vs. 40.1%), consistent with the overall pattern. This suggests that overcorrection is not driven solely by explicit language models. Model capacity itself may contribute through implicit linguistic biases learned from large-scale training data. For CAPT, the practical implication is that better ASR as measured by WER may produce worse pronunciation feedback regardless of architecture.

5.3 Direct Phoneme Comparison

We also examined whether the phoneme recognizer alone, without ASR, can perform mispronunciation detection. We compared G2P-converted canonical text directly against the phoneme recognizer output using PanPhon distance, sweeping the detection threshold to find the oracle operating point. At the best threshold, this baseline achieves FAR of 15.8% but FRR of 67.4% (Table 1). This means that the system catches most mispronunciations but flags two-thirds of correctly pronounced words as errors. Because the phoneme recognizer has a $\sim 15\%$ error rate on L2 speech, it is too noisy for word-level detection on its own. Our method combines both signals: ASR provides word-level candidates and the phoneme recognizer selects among them.

5.4 Severity-Stratified Analysis

Tables 2–4 present detailed analyses for Parakeet, which shows the larger reranking gain; Whisper’s limited N-best diversity (§6.3) produces smaller improvements that do not yield informative subgroup patterns. Table 2 shows the results broken down by error severity on both datasets. On L2-ARCTIC, the improvement scales with severity. The differences were 3.1pp for acceptable-level variations, 4.3pp for moderate errors, and 6pp for severe errors. The high residual FAR on acceptable variations (92.9%) is expected and arguably desirable, since accent features like dental fricative simplification do not impair intelligibility and need not be flagged in most educational contexts.

On SO762, a different pattern emerges. Severe errors (phone accuracy below 1.0) have a much lower baseline FAR (21.5%) than moderate errors (81.9%). Pronunciations that are very different from any recognizable phone are so acoustically distinct that even a strong language model cannot map them back to the intended word. Moderate

Table 2: FAR (%) by error severity (Parakeet). L2-ARCTIC categories follow annotation guidelines; SO762 categories are based on minimum phone accuracy scores.

	L2-ARCTIC		
	Accept.	Moderate	Severe
Baseline	96.0	88.4	80.8
Ours	92.9	84.1	74.8
Δ	−3.1	−4.3	−6.0

	SO762	
	Moderate	Severe
Baseline	81.9	21.5
Ours	76.7	19.0
Δ	−5.2	−2.4

errors, by contrast, are subtle enough for the language model to override the acoustic evidence and produce the canonical word. This suggests that intent bias may be most problematic precisely for the errors that are hardest to detect by other means.

5.5 Learner Demographics: Age Groups

Table 3 presents results stratified by speaker age on the SO762 test split ($n=2,497$ utterances). Children show the largest FAR reduction (6.8pp), followed by teens (5.1pp) and adults (5.0pp). This pattern may reflect greater acoustic variability in younger speakers’ speech, where the language model’s intent-oriented bias is most pronounced. The consistency of improvement across all age groups is relevant for educational deployment, where CAPT systems must serve diverse learner populations without per-demographic tuning. FRR rises by 3.8–4.8pp across strata; the modest cost is within bootstrap CI overlap and consistent with the overall trade-off in Table 1.

Table 3: FAR / FRR (%) by speaker age on SO762 test (Parakeet, $n=2,497$). *Ours* cells show the reranked value with Δ vs. Base.

Age	n	FAR (%)		FRR (%)	
		Base	Ours (Δ)	Base	Ours (Δ)
Child	1,039	67.3	60.5 (−6.8)	11.1	14.9 (+3.9)
Teen	240	84.4	79.3 (−5.1)	7.2	12.0 (+4.8)
Adult	1,218	70.6	65.6 (−5.0)	6.9	10.7 (+3.8)
All	2,497	70.6	65.0 (−5.6)	8.4	12.3 (+3.9)

5.6 L1 Background Generalization

Table 4 shows that the method improves detection across all six L1 backgrounds in L2-ARCTIC,

with FAR reductions ranging from 3.8pp to 4.9pp. Korean and Vietnamese speakers show the largest and smallest baseline FAR respectively (96.0% vs. 81.5%), possibly reflecting differences in how well the ASR language model can normalize each L1’s characteristic error patterns. FRR rises by 1.1–3.1pp across L1 strata; Vietnamese has the smallest FRR cost (1.1pp), Chinese the largest (3.1pp). The absence of any L1 group showing degradation suggests that the method’s reliance on universal articulatory features, rather than L1-specific error models, provides robust cross-linguistic generalization.

Table 4: L1 generalization on L2-ARCTIC (Parakeet, all severity levels). *Ours* cells show the reranked value with Δ vs. Base.

L1	n_{misp}	FAR (%)		FRR (%)	
		Base	Ours (Δ)	Base	Ours (Δ)
Arabic	1,406	89.9	85.5 (−4.4)	5.1	7.7 (+2.6)
Chinese	2,250	91.0	87.2 (−3.8)	5.9	8.9 (+3.1)
Hindi	1,984	95.6	91.3 (−4.2)	3.1	5.5 (+2.4)
Korean	1,236	96.0	91.2 (−4.9)	3.1	6.2 (+3.0)
Spanish	2,451	90.8	86.4 (−4.4)	5.3	8.1 (+2.9)
Vietnamese	4,138	81.5	76.9 (−4.6)	4.0	5.1 (+1.1)

5.7 Reranking Strategy Comparison

Table 5 compares alternative reranking strategies on L2-ARCTIC. PanPhon and the *phoneme-recognizer log-posterior* (PR Log-Posterior) use the same phoneme recognizer in different ways: PanPhon measures the articulatory distance between the hypothesis IPA and the recognizer’s argmax phone sequence, while PR Log-Posterior scores the hypothesis IPA against the recognizer’s frame-wise posterior using a proportional frame alignment. Neither signal is Parakeet’s ASR decoder score, and PR Log-Posterior is not a conventional acoustic-model likelihood. On L2-ARCTIC severe errors, the two signals achieve comparable FAR (74.6 vs. 74.8), but PR Log-Posterior incurs substantially higher FRR (10.5 vs. 7.6) and WER (15.6 vs. 11.8). On SO762 test, PR Log-Posterior wins on FAR (McNemar exact OR = 0.78, 95% CI [0.68, 0.90], $p < 0.001$), but PanPhon remains better on FRR and WER (12.3 vs. 15.9 FRR; 24.9 vs. 28.3 WER). We adopt PanPhon because its CAPT trade-off is more favorable, it avoids the proportional frame alignment that PR Log-Posterior depends on, and its failure modes are more transparent through the surface-anchor audit (§6.2). PanPhon is therefore a Pareto-efficient operating point

on the WER and FRR axes.

Replacing PanPhon with unweighted Levenshtein edit distance on the same IPA sequences yields 2.9pp less FAR reduction (77.7% vs. 74.8%), confirming that articulatory feature weighting meaningfully contributes to the reranking quality. Equal-weight combination of ASR and phoneme scores yields intermediate results. The WER Oracle row shows what happens when the hypothesis closest to the canonical text is selected as an output (i.e., optimized with WER). FAR *increases* to 86.1%, indicating that optimizing for conventional ASR accuracy works against mispronunciation detection. The MDD Oracle shows the best achievable FAR from the N-best list (56.9%), representing the ceiling imposed by beam search pruning.

Table 5: Reranking strategies on L2-ARCTIC severe errors (Parakeet, $n=4,183$). PR Log-Posterior scores each hypothesis under the phoneme recognizer’s posterior, not an ASR decoder likelihood (see §5.7).

Method	WER	FAR	FRR
WER Oracle	5.1	86.1	2.7
ASR Top-1	8.0	80.8	5.9
Levenshtein	9.5	77.7	6.3
Equal Weight	10.1	76.3	6.4
PanPhon (Ours)	11.8	74.8	7.6
Log-Likelihood	15.6	74.6	10.5
MDD Oracle	13.8	56.9	5.4

5.8 N-best List Size

Increasing K from 1 to 5 reduces FAR by 3.6pp (80.8%→77.2%), and extending to $K=10$ provides an additional 2.4pp improvement (77.2%→74.8%), with FRR rising modestly from 5.9% to 7.6%. We use $K=10$ throughout as a practical trade-off.

6 Analysis and Discussion

6.1 Quantifying Intent Bias

To understand the relationship between ASR confidence and overcorrection, we examined how overcorrection rates vary across N-best ranks on SO762 (Table 6). Rank 1, the hypothesis the system is most confident about, has the highest overcorrection rate (69.4%), with the rate decreasing at lower ranks. This is consistent with language model’s preference for canonical forms being strongest at the top of the N-best list, and suggests that surface-faithful alternatives exist at lower ranks but are systematically suppressed.

This finding also explains why word-level confidence thresholding (i.e., flagging low-confidence

Table 6: Overcorrection by N-best rank on SO762 (Parakeet). % Utts = utterances with at least one overcorrection.

Rank	Overcorrection	% Utts
1 (highest)	69.4%	82.0%
3	62.5%	78.3%
10 (lowest)	59.0%	74.7%

words as mispronounced) is not an effective detection strategy. At its oracle operating point on SO762, confidence thresholding achieves only 50.4% FAR with 35.0% FRR (Table 1), which is worse than all other methods. Confidence reflects the language model’s certainty about the *intended* word, not the acoustic fidelity of the transcription.

6.2 Reliability of the Surface Anchor

Our reranker selects the hypothesis whose IPA transcription is closest to a fixed reference: the G2P transcription of the prompt, which we call the *surface anchor*. We write $d_{\text{canonical}}$ for the PanPhon distance from a candidate hypothesis to this anchor. If the anchor itself disagrees with what a correctly-pronouncing learner produces, the reranker has nothing to recover from, and the residual FRR in Table 1 reflects anchor noise rather than reranker error. We test this directly.

We isolate anchor noise on L2-ARCTIC utterances annotated as fully correct ($n=1,083$), where the canonical anchor should agree with the phoneme-recognizer output. Even in this no-error stratum, the median normalized PanPhon distance between anchor and recognizer is 0.52 (IQR=0.49), nearly matching the error-bearing stratum’s median of 0.57. The anchor is therefore systematically noisy across utterances, not selectively noisy for mispronounced speech.

Two stratifications show that this noise propagates into false rejection. First, word-level FRR scales monotonically with anchor distance: from 3.2% in the lowest $d_{\text{canonical}}$ quartile to 11.7% in the highest. Second, comparing utterances by whether their top-1 hypothesis contains an inserted word, the reranker’s helped-to-hurt ratio — newly correct accepts versus newly incorrect rejects — drops from 4.7:1 without insertions to 1.5:1 with insertions (Fisher exact OR=3.04, 95% CI [1.70, 5.46], $p < 0.001$). We therefore suggest single-pronunciation G2P as a measured limitation, and motivate multi-pronunciation lattices as the principled remedy (§6.5).

6.3 ASR Architecture and N-best Diversity

Whisper and Parakeet achieve similar baseline FAR on L2-ARCTIC (82.7% vs. 80.8%), yet Parakeet benefits substantially more from reranking (Table 1). The explanation lies in N-best diversity. Parakeet’s hybrid decoding produces an average of 10 unique hypotheses per utterance, while Whisper’s stronger autoregressive language model constrains beam search to only 4.6 unique hypotheses. In 82.7% of Whisper utterances, the top-1 hypothesis is already the closest to the oracle, leaving little room for reranking to improve. For CAPT applications, this suggests that ASR systems with weaker language models may be preferable when paired with surface-faithful reranking, since they preserve more diverse candidates.

Increasing candidate diversity through generic Whisper decoding does not close this gap: doubling the beam to $K=20$ raises Whisper’s unique-hypotheses-per-utterance only from 2.8 to 4.3 and yields a reranked FAR improvement of just 1.4pp on SO762 test, capturing only 30% of the oracle headroom (4.6pp). The added candidates are mostly linguistic variants rather than acoustically faithful alternatives. The bottleneck is candidate generation under an intent-oriented decoder, not search width; pronunciation-aware decoding (§6.5) is the principled extension.

6.4 Deployment and Educational Implications

An alternative to reranking would be to use a CTC-only model without a language model, which exhibits lower overcorrection (Figure 2). However, such models produce transcripts with 46–50% word error rate, making them unsuitable for learner display. Our approach preserves the benefits of a strong ASR system while mitigating its overcorrection: a single ASR pass produces two outputs, where the top-1 hypothesis provides a readable transcript for display while the reranked hypothesis drives mispronunciation detection internally. Reranking adds negligible compute beyond N-best extraction.

The FAR/FRR distinction maps onto two failure modes in pronunciation feedback: high FAR leaves errors undetected, while high FRR frustrates learners with false alarms (Neri et al., 2002). Our method’s profile—lower FAR with a modest FRR rise—is best suited to intermediate and advanced learners; the age-group results (Table 3) suggest particular benefit for younger learners.

Table 7: Examples from L2-ARCTIC. The top-1 hypothesis corrects the learner’s pronunciation and the reranked hypothesis preserves the surface form.

Type	Top-1 (intent)	Reranked (surface)
Omission	... she is coming...	... she coming...
Subst.	... previous wives...	... previous wise...
Acoustic	... was stream- ing ...	... were trimming...

6.5 Future Work

WER alone is insufficient for evaluating ASR in CAPT (Figure 2); we suggest the *Pronunciation Preservation Rate* ($PPR = 1 - FAR$) as a complementary metric (range 25.8%–59.0% on SO762 test). Four directions remain open. (i) **Multi-pronunciation lattices** replacing the single canonical G2P, motivated by the surface-anchor audit (§6.2) showing $3.7\times$ higher FR rate in the top $d_{\text{canonical}}$ quartile—a likely route to reducing the dominant FRR cost. (ii) **Latency-aware two-pass diagnosis**, running a wider diagnostic beam ($K=100\text{--}200$) alongside the display ASR; better offline detection at the cost of user-facing latency. (iii) **Pronunciation-aware decoding** objectives that balance LM probability against acoustic faithfulness, motivated by the §6.3 finding that wider beams ($K=20$) do not recover surface forms. (iv) **Learning-outcome user studies** to verify that the 5.6pp absolute / 19% relative recall gain translates to learning benefit.

6.6 Qualitative Examples

Table 7 illustrates three common patterns where the top-1 hypothesis masks a pronunciation error that the reranked hypothesis exposes. In the first example, the learner omits the verb, a common pattern for speakers whose L1 lacks obligatory verb marking. Here, the ASR inserts “is” because it is grammatically expected. In the second, the learner produces a word closer to “wise” than “wives,” but the language model selects the contextually appropriate word. A more complex case appears in the third example, where multiple phonetic confusions lead to a substantially different surface form.

The method can also introduce errors. The 1.7–3.0pp FRR increase (Table 1) occurs when the phoneme recognizer output happens to be closer to an incorrect hypothesis than to the canonical text, and it often happens for short function words (e.g., “a,” “the”) where phoneme sequences are

too brief for PanPhon distance to discriminate reliably. These false alarms are concentrated in high-frequency, low-information words that are less relevant for pronunciation feedback.

7 Conclusion

The results of the present study suggest that WER alone is insufficient for evaluating ASR systems intended for pronunciation training. Across eight systems spanning three architectures, lower WER consistently predicts higher overcorrection of mispronunciations. We propose Pronunciation Preservation Rate as a complementary metric that captures how well an ASR system preserves learner errors in its output.

Surface-faithful reranking provides a practical mitigation. By selecting N-best hypotheses that are phonetically closer to what was actually pronounced, the method reduces false acceptance of mispronunciations by 6.0 and 5.6pp on two datasets, across diverse age groups and L1 backgrounds, without retraining or access to ASR internals.

The method’s ceiling is bounded by beam-search pruning, and direct phoneme comparison alone produces too many false alarms; pronunciation-aware decoding (§6.5) is the principled extension.

Limitations

Several limitations remain. (1) Residual FAR is high (74.8% on L2-ARCTIC severe), and the $K=10$ oracle ceiling (56.9%) shows the N-best list lacks surface-faithful hypotheses for many utterances—a beam-search limitation. (2) We evaluate on read speech with known prompts; spontaneous L2 speech would require a different alignment formulation. (3) Our evaluation is limited to L2 English, and the G2P component is language-specific. (4) The phoneme recognizer has a 19.2% error rate on L2-ARCTIC (15.0%–24.5% across L1s), which propagates into reranking. (5) The single-canonical-G2P anchor is an approximation: §6.2 shows it disagrees with the recognizer on correctly-pronounced words and predicts false rejection; multi-pronunciation lattices (§6.5) are the principled fix. (6) We have not conducted user studies; the 5.6pp absolute / 19% relative recall gain is a necessary but not sufficient indicator of learning benefit.

Acknowledgments

We thank the BEA 2026 reviewers for their constructive feedback, which led to the additional analyses and clarifications presented in this revision.

References

- Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA. Curran Associates Inc.
- Enrique Cámara-Arenas, Cristian Tejedor-García, Carlos J. Tomás-Vázquez, and David Escudero-Mancebo. 2023. Automatic pronunciation assessment vs. automatic speech recognition: A study of conflicting conditions for L2-English. *Language Learning & Technology*, 27(1):1–19.
- Chen Chen, Ruizhe Li, Yuchen Hu, Sabato Marco Siniscalchi, Pin-Yu Chen, EngSiong Chng, and Chao-Han Huck Yang. 2024. It’s never too late: Fusing acoustic information into large language models for automatic speech recognition. In *The Twelfth International Conference on Learning Representations*.
- Maxine Eskenazi. 2009. An overview of spoken language technology for education. *Speech Communication*, 51(10):832–844. Spoken Language Technology for Education.
- Yuan Gong, Ziyi Chen, Iek-Heng Chu, Peng Chang, and James Glass. 2022. Transformer-based multi-aspect multi-granularity non-native English speaker pronunciation assessment. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7262–7266.
- Daniel Korzekwa, Jaime Lorenzo-Trueba, Szymon Zaporowski, Shira Calamaro, Thomas Drugman, and Bozena Kostek. 2021. Mispronunciation detection in non-native (12) English with uncertainty modeling. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7738–7742.
- Wai-Kim Leung, Xunying Liu, and Helen Meng. 2019. CNN-RNN-CTC based end-to-end mispronunciation detection and diagnosis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8132–8136.
- Kun Li, Xiaojun Qian, and Helen Meng. 2017. Mispronunciation detection and diagnosis in 12 English speech using multidistribution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):193–207.
- Somshubra Majumdar and Nithin Rao Koluguri. 2024. Pushing the boundaries of speech recognition with NVIDIA NeMo parakeet ASR models. NVIDIA Technical Blog.
- David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. Pan-Phon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee.
- Ambra Neri, Catia Cucchiaroni, Helmer Strik, and Lou Boves. 2002. The pedagogy-technology interface in computer assisted pronunciation training. *Computer Assisted Language Learning*, 15(5):441–467.
- Kyubyong Park and Jongseok Kim. 2019. g2pE. <https://github.com/Kyubyong/g2p>.
- Linkai Peng, Kaiqi Fu, Binghuai Lin, Dengfeng Ke, and Jinsong Zhan. 2021. A Study on Fine-Tuning wav2vec2.0 Model for the Task of Mispronunciation Detection and Diagnosis. In *Interspeech 2021*, pages 4448–4452.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrina Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.
- SYSTRAN. 2023. faster-whisper. <https://github.com/SYSTRAN/faster-whisper>.
- Silke M. Witt and Steve J. Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication*, 30(2–3):95–108.
- Yongkook Won. 2025. Assessing the efficacy of word error rate as a proxy for pronunciation quality. *Journal of Second Language Pronunciation*, 11(3):394–422.
- Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, and Long Ma. 2021. Explore wav2vec 2.0 for Mispronunciation Detection. In *Interspeech 2021*, pages 4428–4432.
- Klaus Zechner, Derrick Higgins, Xiaoming Xi, and David M. Williamson. 2009. Automatic scoring of non-native spontaneous speech in tests of spoken English. *Speech Communication*, 51(10):883–895.
- Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yukai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. speechocean762: An Open-Source Non-Native English Speech Corpus for Pronunciation Assessment. In *Interspeech 2021*, pages 3710–3714.

Guanlong Zhao, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. 2018. [L2-ARCTIC: A Non-native English Speech Corpus](#). In *Interspeech 2018*, pages 2783–2787.

KEYSCORE — Keystroke-enhanced Automated Essay Scoring

Nils-Jonathan Schaller¹, Daniel Mora¹,
Thorben Jansen¹, Olaf Köller¹, Andrea Horbach^{1,2}

¹Leibniz Institute for Science and Mathematics Education, Kiel, Germany

²Kiel University, Kiel, Germany
schaller@leibniz-ipn.de

Abstract

We investigate the predictive power of keystroke logging data for automated essay scoring using the newly collected PISA FLA writing process dataset. Based on 3,882 writing sessions, we extract a comprehensive set of keystroke-based process features, including temporal measures, pause and burst patterns, deletion behavior, production efficiency, and navigation activity and evaluate their ability to predict holistic essay scores on a 0–5 scale. We specifically compare process-feature-based models with content-based scoring approaches trained on data written with and without the help of an AI chatbot, and investigate how predictive power evolves over the course of a writing session by training models at multiple time thresholds. Our analysis reveals that keystroke features provide genuine early predictive signal, capturing aspects of writing fluency and revision behavior that distinguish writers before their texts are long enough to score conventionally. Additionally, our results suggest that process-based scoring is a viable complement to product-based approaches, with promise for formative, real-time feedback during writing.

1 Introduction

Feedback is most effective when it is timely and targets the writing process rather than only the finished product (Hattie and Timperley, 2007; Graham et al., 2015; Shute, 2008). However, much of the traditional automated assessment literature focuses on evaluating completed texts and their summative outcomes rather than providing in-process feedback (Cotos, 2015).

One way to achieve such feedback is by constant monitoring of the writing process, which is often hindered by a lack of suitable writing process data. To address this research gap, we present a new dataset PISA FLA, consisting of almost 4000 English-as-a-foreign-language essay writing sessions from German participants in the 2025 PISA

data collection. In this dataset, students responded to two individual argumentative writing prompts and produced texts under two conditions: with and without the assistance of a generative AI chatbot (hereafter referred to as *Chat* and *NoChat*, respectively). This dataset thus offers ample opportunities to investigate both how students write and how essays can be automatically scored already during the writing process.

We investigate the latter question in three different setups: By means of traditional transformer-based essay scoring models, through keylog-based process features and by combining the two approaches. We further address the question of how early on (within the 30 minutes of a single writing session) reliable summative feedback can be given, i.e. feedback with an accuracy similar to that of a model trained on full texts. We achieve this by comparing performance at certain points in time of the writing session. Finally, we investigate the influence of chatbot support on automated scoring.

In doing so, our paper makes the following contributions:

- We present a new dataset of EFL essays together with detailed keylog information making the writing process transparent.
- We show that reliable automated feedback does not require a complete text, but that there is a tradeoff between essay scoring performance and writing time.
- We show that integrating process features into a transformer-based model improves prediction at early writing stages.

The data is available by request until September 2027 after which it will be openly published: https://github.com/darius-ipn/keystroke_pisa2025_fla.

After 5 minutes
<i>Disagree:</i> – more important that the students learn sth about the subject – the teacher doesnt have to be in a good relationship with the students, the teacher only has to be polite – the students dont learn anything if the teacher is polite
After 10 minutes
<i>Disagree:</i> – more important that the students learn sth about the subject – the teacher doesnt have to be in a good relationship with the students, the teacher only has to be polite – the students dont learn anything if the teacher is polite but doesnt know sth about the topic <i>Argumentation:</i> Today we are talking about the question if a teacher's ability to relate well with students is more important than excellent knowledge of the subject being taught.

Figure 1: Writing snapshots from a single participant. After 5 minutes, the student has only produced planning notes in bullet-point form. By 10 minutes, an introductory sentence has been added, illustrating the transition from planning to text production.

2 Related Work

Our work builds on two lines of research: keystroke logging as a window into writing processes, and automated essay scoring. We review relevant work in each area before describing how our approach connects them.

2.1 Writing Process Research and Keystroke Logging

Writing research has long established that composing involves planning, producing text, and revising, and that writers switch between these processes rather than completing them in a fixed order (Flower and Hayes, 1981). Later models distinguish between the cognitive effort of formulating ideas and typing them out (Hayes, 2012). Keystroke logging offers a way to observe these processes (Leijten and Van Waes, 2013).

Recent work has demonstrated that providing students with keystroke-based feedback on their writing process can improve text quality (Vandermeulen et al., 2020). Allen et al. (2016) showed that keystroke indices, particularly total number of keystrokes, explain a large proportion ($R^2 = .76$)

of variance in essay scores for L1 English undergraduates writing in an intelligent tutoring system. However, Conijn et al. (2022) investigated which keystroke features are useful for predicting writing quality during the writing process but found the relationship to be limited. They did show that feature importance shifts over the course of writing, an observation our work further addresses at larger scale and with complementary text-based models.

Several keystroke corpora with quality ratings have recently become available, including KLiCKe by Tian et al. (2025) with $\sim 5,000$ argumentative essays from L1/L2 adults and KUPA-KEYS by Velentzas et al. (2024) with 1,006 EFL essays graded on the CEFR scale. Our dataset complements these resources by focusing on adolescent EFL learners in a standardised assessment context and by including an AI-assisted writing condition.

2.2 Automatic Essay Scoring

Recent work on automated essay scoring has moved toward transformer-based architectures (Li and Ng, 2024). DeBERTa has shown strong performance in this context, outperforming BERT and RoBERTa on the ASAP dataset (Susanto et al., 2024) and serving as the model of choice in top entries of the Kaggle Automated Essay Scoring 2.0 competition, where winning solutions used ensembles of DeBERTa-v3-large with two-staged training (Learning Agency Lab, 2024). Based on these results, we adopt DeBERTa-v3-base as the text encoder in our experiments.

However, many transformer-based AES systems rely solely on the learned text representation for scoring. Uto et al. (2020) demonstrated that concatenating essay-level features (e.g., readability indices, syntactic counts) with a transformer’s [CLS] representation before the scoring layer yields consistent gains, with minimal additional parameters. Building on this, Lohmann et al. (2025) systematically compared feature-based, embedding-based, and hybrid models for analytic trait scoring on argumentative essays from both L1 and L2 learners, including essays from the MEWS corpus (Keller et al., 2020) that shares our writing prompts. Their hybrid architecture, combining 220 linguistic features with DistilBERT embeddings, consistently outperformed the feature-only and embedding-only models, with the largest gains for content and organization assessment.

Additionally, they showed that linguistic features and contextual embeddings capture partially differ-

ent aspects of text quality, supporting architectures that combine both sources. More broadly, [Li and Ng \(2024\)](#) argue that AES research should move beyond optimizing scores on standard benchmarks toward new task settings and formative applications — a direction our work pursues by evaluating scoring across writing time thresholds.

Our work extends this line of research in two directions. To our knowledge, no prior work has integrated keystroke process features directly into a transformer’s prediction architecture.

First, we replace the linguistic feature vector with keystroke process features, testing whether signals from the writing process can complement a transformer’s text representation in the same way that linguistic features do. Second, we evaluate this combined model not only on completed essays but across multiple time thresholds during writing, addressing the question of how early in the writing process reliable scoring becomes feasible.

3 Data

In the following we present the PISA 2025 FLA dataset used in our experiments. The dataset was collected in the 2025 PISA assessment as an additional task for foreign language assessment in German High schools on students at the age of 15. It comprises initially 4,091 writing sessions, in which students responded to argumentative writing prompts in English as a foreign language. Students were assigned to one out of two groups: Those that had access to an AI chatbot for writing support and those without. The chatbot could be used unconditional for all kinds of prompts. The underlying model was gpt-4o-mini. After data cleaning (see Section 3.1) we are left with 3,882 essays, which are distributed roughly evenly across the two conditions, as displayed in Table 1. The two conditions differ substantially in writing outcomes. Students with chat access produced considerably longer texts (1,813 vs. 1,138 characters on average; 274 vs. 200 words) and received higher holistic scores (mean 3.2 vs. 2.1 on a 0–5 scale), with 20.2% of chat-assisted essays reaching the maximum score of 5 compared to just 2.4% without chat. Notably, chat-assisted students achieved these longer, higher-scored texts with fewer keystrokes (1,666 vs. 1,951), suggesting substantial use of copy-pasting from the chatbot. The writing was limited to 30 minutes for all conditions although the majority stopped at approximately 25 minutes.

	<i>NoChat</i>	<i>Chat</i>
N (scored)	1,911	1,971
Avg. text length (chars)	1,138	1,813
Avg. text length (words)	200	274
Avg. sentences	8.9	14.0
Score (mean)	2.1	3.2
Score (median)	2.2	3.2
% scoring 5	2.4	20.2
Avg. writing time (min)	24.7	25.2
Avg. active writing ratio	0.5	0.4
Avg. keystrokes	1,951	1,666

Table 1: Descriptive statistics by condition. Scores are on a 0–5 holistic scale. Active writing ratio is 1 minus the proportion of break time (pauses ≥ 2 s) relative to total session time.

The active writing ratio was slightly higher for *NoChat* (0.5 vs. 0.4), consistent with *Chat* students spending time interacting with the chatbot interface rather than typing continuously.

3.1 Data Cleanup

To ensure data quality, we applied two filtering criteria: (1) submissions with fewer than 10 characters were excluded as empty or trivial responses, and (2) submissions exceeding 10,000 characters (above the 99th percentile) were excluded as they reflected excessive copy-pasting behaviour rather than authentic writing processes.

Of the initial $N = 4,091$ submissions, 130 were excluded for insufficient length and 35 for excessive length, resulting in a sample of $N = 3,926$ (96.0% retention). Of these, 3,882 had valid holistic scores and are the final analytical sample.

3.2 Keystroke Measures

Keystrokes and texts were collected within a self-developed web interface. From each writing session we extract 25 process features covering six behavioural dimensions. **Temporal measures** capture total writing time (first to last keystroke) and initial pause (time from task start to first keystroke). **Pause and break patterns** characterize interruptions in typing: we count all pauses $\geq 2,000$ ms as breaks and derive their total duration, mean duration, and the ratio of break time to total writing time. **Burst characteristics** describe the production sequences between breaks or revisions, including burst count, mean burst length (in characters), and mean burst duration. **Deletion behaviour** is captured through the total number of delete/backspace keystrokes, the deletion ratio (deletions / total keystrokes) and total characters deleted. **Produc-**

tion efficiency includes final text length (characters and words), characters per minute (relative to active writing time), total keystrokes, and the process-to-product ratio (total characters typed / final text length), which indicates how much text was produced and subsequently removed. Finally, **navigation activity** is measured through copy-paste count, a linearity index reflecting the proportion of forward cursor transitions (1 = fully sequential writing), and area switches between interface regions. For sessions in the *Chat* condition, we additionally capture time allocation across four interface areas (text editor, task text, chat window, chat prompt) and the number of area switches; these features are set to zero for NoChat. All features are z-score normalised within each time threshold (see Section 3.4), so that distributions reflect the writing stage rather than the completed essay. See also Appendix A, Table 7.

The features were selected to cover established behavioral dimensions in keystroke logging research such as temporal, pause, burst, deletion, production, and navigation measures. Note that the full dataset also tracks mouse movements and other non-keyboard events, which we partially integrate such as cursor reposition events, paste events, interface area switches, and time in chat/task/editor/no area.

3.3 Dataset Annotation

In the following we describe the automatic and manual annotation efforts to obtain the holistic scores used in our scoring experiments.

3.3.1 Creating a silver standard

For our current study, we were not able to obtain human annotations for the full dataset. As a proxy for a human gold standard, we employ an automated scoring model which has been trained on essays from the MEWS dataset, i.e., essays answering the same writing task but from a slightly different learner population.

Specifically, a linear regression model was trained on the MEWS dataset (Keller et al., 2020, 2024) to predict both holistic scores and individual trait scores. The model uses 220 linguistic features extracted in the same way as in Lohmann et al. (2025). These features capture various aspects of the text, including lexical variety, syntactic structure, cohesion, as well as error-, frequency- and length-based indicators. Before training, all features are scaled using a min-max scaler, and

multicollinearity is reduced by removing highly correlated features (correlation > 0.9).

This silver standard is only used as training material. All models in this study are evaluated against the smaller human-annotated test dataset described in the following, ensuring that reported results reflect agreement with human judgments.

3.3.2 Gold Standard Scores

To obtain a gold standard test set, we drew a stratified random sample from the corpus. Strata were defined by domain (Teacher vs. Advertisement), experimental condition (*Chat/NoChat*), and a quality profile constructed from tertile bins (Low / Medium / High) of silver-standard analytic scores for Content, Organization, and Language Quality, yielding up to 27 performance clusters per domain-condition cell. Allocation was proportional to population frequency with a minimum-count floor to retain rare profiles.

The final sample comprises 429 essays (212 *NoChat*, 217 *Chat*), each double-annotated by trained raters (Krippendorff’s $\alpha = .79$).

Four student annotators with a background in education were trained on the original MEWS scoring rubrics. Each essay was randomly assigned and annotated twice. Inter-annotator agreement reached a Krippendorff’s α of 0.79 (interval metric), indicating substantial agreement. Final gold scores were obtained by averaging.

These annotations were also used to validate the silver standard: We evaluated the automatic scoring model (trained on MEWS data) against the gold annotations and obtained a QWK of 0.76 between the automatically annotated silver standard and the human gold standard. This is comparable to the inter-annotator agreement, supporting the use of the silver labels as training signal.

The gold-annotated subset serves as the held-out test set in all experiments.

3.4 Time Threshold Splits on Test Set

One core research question we aim to answer is: How early in the writing process can we give meaningful feedback to learners? To this end, we reconstruct from the keylog data for each essay its state after 5, 10, 15, 20 and 25 minutes of writing time, reflecting planning notes, partial drafts, or incomplete revisions rather than a finished product (see Figure 1 for an illustration). A small number of writing sessions (1–2 per threshold) with no keystroke activity up to that time, i.e. still empty

texts, were excluded. Note that our goal is to judge the assumed quality of the finished essay early in the writing process. Therefore we do not collect new human judgements for these incomplete texts, but propagate the label from the full essay to all earlier stages of that essay.

4 Method

In the following, we describe first the different model infrastructures for scoring, then detail the shared experimental setup covering data splits, time thresholds, and evaluation.

4.1 Scoring Models

We approach in-process essay scoring through three complementary model families: (1) process-feature models that predict scores exclusively from keystroke behaviour, without access to the essay text; (2) transformer-based models that rely on the essay text; and (3) an architecture that combines both information sources. All models are trained and evaluated separately for *Chat* and *NoChat* and at six time thresholds throughout the 30-minute writing session, allowing us to trace how prediction quality evolves over the course of composition.

4.1.1 Process-Feature Baselines

To evaluate the predictive power of keystroke features independently of essay content, we train gradient-boosted regression trees (XGBoost) that operate exclusively on the process features described in Section 3.2. Each model is trained in one of three feature configurations: (1) *all* keystroke features, (2) *length only*, using only the text length in characters at the respective time threshold as a single predictor, and (3) *no length*, using all keystroke features except text length. This decomposition disentangles the contribution of text length, a known dominant predictor in automated essay scoring (Allen et al., 2016), from the behavioural signal captured by the remaining process features such as pause patterns, burst characteristics, and deletion behaviour. Hyperparameters are selected via grid search over the number of estimators (100, 200, 500), maximum tree depth (3, 5, 7), learning rate (0.01, 0.05, 0.1), and L1 regularisation strength (0, 0.1, 1.0), with subsampling rate and column sampling fixed at 0.8. The combination maximising QWK on the dev set is selected. We also trained ordinary least squares regression models in the same configurations; results are reported in Appendix A Table 6.

4.1.2 Text-Based Scoring with DeBERTa

We fine-tune DeBERTa-v3-base (He et al., 2021) with a single-output regression head predicting the holistic score on a continuous 0–5 scale. Input texts are tokenised with the model’s default tokeniser. Due to hardware restrictions it was necessary to truncate essays to a maximum of 512 tokens. In the training set, this led to truncation in 8.0% of texts, with truncated samples losing on average 24% of their tokens. At intermediate timepoints, truncation was less frequent, as texts were naturally shorter.

Training uses AdamW with a learning rate of 2×10^{-5} and a batch size of 8 for up to 10 epochs with early stopping.

We evaluate two training variants:

Full-text model. The model is trained exclusively on complete essays. At inference, it receives either the full essay or an intermediate text snapshot at a given time threshold. This reveals how well a model trained on finished writing generalises to incomplete drafts.

Hybrid model. The model is trained on the hybrid training set described in Section 4.2.1, which combines complete essays with intermediate snapshots, exposing the model to both complete and in-progress texts.

4.1.3 Combined Text and Process Features

Following Uto et al. (2020), who concatenated handcrafted linguistic features with a transformer’s text representation, we inject keystroke process features into the DeBERTa prediction head. We concatenate DeBERTa’s pooled text representation with a keystroke feature vector and train a joint regression head on both information sources simultaneously. Concretely, we obtain the pooled CLS representation from DeBERTa-v3-base, a 768-dimensional vector produced by the context pooler, which applies a learned linear projection and activation on top of the [CLS] token. This vector is concatenated with a 32-dimensional side vector composed of two parts: (1) the 25 z-scored keystroke process features described in Section 3.2, and (2) a 6-dimensional one-hot encoding of the source timepoint (*5 min*, *10 min*, *15 min*, *20 min*, *25 min*, *full*). The timepoint indicator is necessary because the keystroke features are z-score normalised *within* each time threshold. Without this signal, the model cannot distinguish between normalisation groups. Because z-score normalisation on small per-timepoint groups can produce

Score	NoChat			Chat		
	Train	Dev	Test	Train	Dev	Test
0.0	134	15	16	60	7	5
0.5	82	9	18	34	4	5
1.0	122	13	18	47	5	10
1.5	205	23	25	94	10	12
2.0	241	27	32	142	16	20
2.5	289	32	36	212	24	18
3.0	217	24	29	218	24	39
3.5	119	13	16	179	20	23
4.0	69	8	10	153	17	23
4.5	17	2	4	119	13	19
5.0	34	4	8	320	36	43
<i>N</i>	1,529	170	212	1,578	176	217

Table 2: Distribution of holistic scores (binned to nearest 0.5) across train/dev/test splits for both conditions.

extreme values, we clip all keystroke z-scores to the range $[-5, +5]$ before training. This prevents rare outliers from destabilising gradient updates, an issue we observed during initial experiments. The combined 800-dimensional vector ($768 + 32$) is passed through a two-layer feed-forward network ($800 \rightarrow 128 \rightarrow 1$) with GeLU activation and dropout ($p=0.1$), producing a continuous regression score. For ordinal metrics (QWK, macro F_1), predictions are clipped to $[0, 5]$ and rounded to the nearest 0.5, yielding 11 categories. The entire model is trained end-to-end with the same hyperparameters as the text-only DeBERTa model.

4.2 Experimental Setup

4.2.1 Data Splits

We use the 429 gold-annotated essays described in Section 3.3.2 as our fixed test set. The remaining data is split at a 90/10 ratio using stratified sampling on holistic scores binned to the nearest 0.5. Rare score bins with fewer than 2 instances are merged into their nearest neighbour before stratification. Table 2 shows the resulting distributions. Scores in the *NoChat* condition concentrate around 2.0–3.0, whereas the *Chat* condition is right-skewed, with the 5.0 bin alone accounting for roughly 20 % of sessions.

Hybrid training set. Following Schaller et al. (2025), we construct a hybrid training set by pairing each full essay with one randomly sampled intermediate snapshot (drawn uniformly from the five time thresholds), approximately doubling the training set from 3,107 to 6,199 instances (see Table 3). This exposes the model to both complete and in-progress texts, reducing its reliance on struc-

tural cues present only in finished essays.

4.2.2 Summary

All models are trained and evaluated at six time thresholds (5, 10, 15, 20, 25 minutes, and full essay). For the process-feature models, we train one XGBoost model per combination of condition (*Chat*, *NoChat*), feature configuration (*all*, *length only*, *no length*), and time threshold, yielding 36 combinations. For the text-based models, we evaluate three DeBERTa-v3-base variants: *full-text*, *hybrid* (Section 4.2.1), and *combined*. All predictions are clipped to $[0, 5]$ and rounded to the nearest 0.5 for QWK computation.

	NoChat	Chat	Total
Full texts	1,529	1,578	3,107
5 min	306 (20.0%)	329 (21.0%)	635
10 min	266 (17.4%)	315 (20.2%)	581
15 min	330 (21.6%)	310 (19.8%)	640
20 min	303 (19.8%)	294 (18.8%)	597
25 min	324 (21.2%)	315 (20.2%)	639
Hybrid total	3,058	3,141	6,199

Table 3: Distribution of training samples in the Hybrid training set. Each essay appears once as its full text and once as a randomly sampled intermediate snapshot.

5 Results

In the following, we first present results on the process-feature-based models, then the text-based transformers and finally the combined architecture.

5.1 Process-Feature-Based Scoring

Table 4 presents results for the three keylog-based feature configurations: all keystroke features (ALL), text length only (LEN), and all features except length (NOLEN), separately for the two experimental conditions (*NoChat*, *Chat*)

Unsurprisingly, prediction quality improves steadily over the course of the writing session, rising from QWK .32 at 5 minutes to .76 at completion for *NoChat* (all features), and from .59 to .91 for *Chat*. The steepest gains occur between 5 and 15 minutes, after which performance improvements level off, suggesting that the first half of the writing session already captures most of the predictive signal. The relative contribution of text length and behavioural process features shifts markedly over time. At 5 minutes, LEN alone is a weak predictor (QWK of .14 and .31), while NOLEN already achieve substantially higher performance (.38 and

.55). This pattern reverses later: by completion, length alone reaches .77 (*NoChat*) and .92 (*Chat*), matching the full feature set. This confirms the well-established role of text length as a dominant predictor in automated essay scoring (Allen et al., 2016) while demonstrating that its dominance only holds once sufficient text has been produced.

Notably, the NOLEN configuration performs competitively with ALL across both conditions, matching it at most time thresholds, with small fluctuations in either direction that likely reflect estimation noise given the test set size. This suggests that explicit text length is largely redundant given the remaining process features.

Model performance was consistently higher in the *Chat* condition, with QWK reaching .91 at completion compared to .76 *NoChat*. This likely reflects two compounding factors: *Chat*-assisted students produced longer, more content-rich texts earlier in the session, providing a stronger signal for scoring models, and the right-skewed score distribution — with 20.2% of essays at the maximum — may additionally increase class separability across all models. A more fine-grained analysis of individual AI usage strategies, including copy-paste behaviour and prompt intent, is left for future work. These results suggest that keystroke process features provide genuine early predictive signal, capturing aspects of typing fluency, planning behaviour, and revision activity that distinguish writers before their texts are long enough to score conventionally. Linear regression models trained in the same configurations showed comparable patterns at lower overall performance (see Appendix 6), suggesting that these findings are robust to model choice.

To further investigate which features drive prediction, Appendix A Table 8 reports Pearson correlations of each feature with text length and holistic score, separately by condition. Text length is the strongest predictor of score ($r = .73$ in both conditions). That the NOLEN XGBoost configuration matches or exceeds ALL at most time thresholds (Table 4) indicates, that the combination of features can match or exceed the length signal as a predictor of score. The correlation differs between conditions: `total_keystrokes` and `break_count` are positive predictors in *NoChat* but show near-zero or negative correlations in *Chat* ($r = -.01$ and $-.14$), likely because higher keystroke counts in *Chat* reflect chatbot interaction rather than productive writing. `initial_pause` is the most condition-stable

predictor ($r_{\text{score}} = -.19$ *NoChat*, $-.16$ *Chat*), indicating that students who begin writing sooner tend to score higher regardless of final text length.

	<i>NoChat</i>			<i>Chat</i>		
	All	Len.	NoLen.	All	Len.	NoLen.
5 min	.32	.14	.38	.59	.31	.55
10 min	.52	.35	.58	.71	.55	.71
15 min	.57	.59	.62	.79	.74	.78
20 min	.65	.64	.68	.88	.86	.86
25 min	.76	.74	.76	.91	.91	.90
FullTx	.76	.77	.75	.91	.92	.91

Table 4: QWK for XGBoost models at each time threshold. ALL = all keystroke features; LEN = text length only; NOLEN = all features except length. Best per condition and threshold in bold.

5.2 Text-Based Scoring

Table 5 reports QWK for the two DeBERTa models (FULL and HYBRID) evaluated at each time threshold and on complete essays, separately for *NoChat* and *Chat*.

	<i>n</i>	Full	Hyb.	Δ_H	Combi.	Δ_C
<i>NoChat</i>						
5 min	212	.15	.33	+19	.43	+28
10 min	212	.42	.53	+11	.63	+21
15 min	212	.65	.63	−.03	.69	+04
20 min	212	.77	.67	−.10	.72	−.05
25 min	212	.86	.74	−.13	.77	−.09
FullTx	212	.86	.75	−.12	.77	−.10
All	1272	.58	.62	+04	.67	+09
<i>Chat</i>						
5 min	217	.21	.40	+19	.58	+37
10 min	217	.41	.60	+19	.73	+32
15 min	217	.62	.73	+11	.83	+21
20 min	217	.84	.86	+02	.90	+05
25 min	217	.95	.90	−.05	.94	−.01
FullTx	217	.95	.90	−.05	.94	−.01
All	1302	.57	.73	+16	.82	+26
<i>Chat+NoChat</i>						
5 min	429	.22	.43	+21	.58	+36
10 min	429	.45	.62	+17	.73	+28
15 min	429	.67	.73	+06	.81	+13
20 min	429	.84	.81	−.03	.85	+01
25 min	429	.92	.85	−.07	.88	−.04
FullTx	429	.93	.86	−.07	.88	−.04
All	2574	.62	.73	+11	.79	+18

Table 5: QWK by time threshold for FULL, HYBRID, and COMBI. DeBERTa models. Δ_H and Δ_C denote differences to FULL.

Full-text model. When trained exclusively on complete essays, DeBERTa achieves strong performance at full length (QWK of .86 and .95 for

the *NoChat* and *Chat* conditions, respectively), outperforming the best XGBoost keystroke-feature models (.77 and .92; Table 4). However, performance degrades sharply on incomplete drafts: at 5 minutes, QWK drops to .15 (*NoChat*) and .21 (*Chat*). This is expected: in the training data, a short, structurally incomplete text is often a weak finished essay, whereas in the test data it may be an early snapshot of an ultimately high-scoring one.

Hybrid model. Hybrid training yields clear improvements at early time thresholds: QWK rises by +.19 at 5 minutes in both conditions, and gains persist through 10 minutes (+.11 / +.19) and, in the *Chat* condition, through 15 minutes (+.11). This is consistent with Schaller et al. (2025): training on a mix of complete and incomplete texts helps the model generalise to intermediate writing stages.

However, the hybrid model incurs a cost at later stages: from 15–20 minutes onward, QWK falls below the full-text model, with the largest gap at 25 minutes in the *NoChat* condition (−.13). The full-essay ceiling is also lower (.75 vs. .86 *NoChat*; .90 vs. .95 *Chat*), indicating that the mixed training signal dilutes the model’s ability to fully exploit the richer information available in complete texts.

Condition differences. The crossover point, where the hybrid model transitions from outperforming to underperforming the full-text model, differs between conditions. For *NoChat*, the hybrid model falls behind already at 15 minutes; for *Chat*, it remains competitive through 20 minutes and only drops at 25 minutes. This asymmetry likely reflects the role of AI chat scaffolding: chat-assisted students produce more structured, content-rich text earlier in the session, providing the hybrid model with more “scorable” material at intermediate time points. This is reinforced by consistent larger hybrid gains at early thresholds in the *Chat* condition.

Comparison with keystroke-feature models. Compared to the XGBoost keystroke-feature models, the DeBERTa models achieve higher QWK once sufficient text is available: at full length, DeBERTa (full) reaches .86/.95 vs. .76/.91 for the best XGBoost configuration. At early stages the picture reverses: the XGBoost all-features model achieves QWK of .59 at 5 minutes in the *Chat* condition, outperforming both DeBERTa (full) at .21 and DeBERTa (hybrid) at .40. In the *NoChat* condition, it outperforms the full-text model (.32 vs .15) but is on par with the hybrid model (.32 vs

.33). This confirms that keystroke and text features capture complementary signals: process features provide more robust early prediction when text is minimal, while text-based models dominate once enough content has been produced.

5.3 Combined Text and Process Feature Scoring

Table 5 extends the comparison with COMBINED, which concatenates DeBERTa text embeddings, keystroke features, and a timepoint one-hot encoding (see also Figure 2).

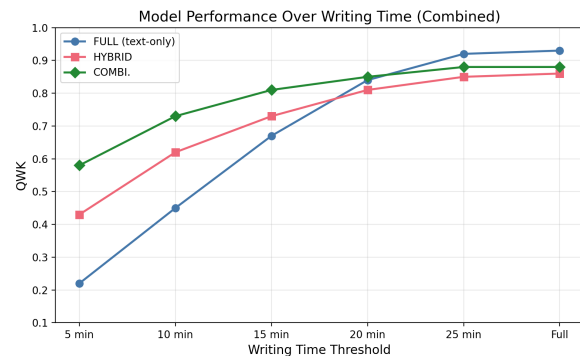


Figure 2: Performance of all 3 DeBERTa models.

Performance across writing stages. At early time thresholds, where essay text is still sparse, COMBINED consistently outperforms both FULL and HYBRID. At 5 minutes, the gains over FULL are substantial (+.28 *NoChat*, +.37 *Chat*), confirming that keystroke process features compensate for what the transformer cannot yet extract from minimal text. Even compared to HYBRID, COMBINED adds +.10 to +.18 QWK at the 5-minute mark. As essays grow longer, the advantage narrows: by 20 minutes in the *NoChat* condition, FULL overtakes COMBINED, and at essay completion, FULL achieves the highest QWK in *NoChat* (.86 vs .77), while in *Chat* the two models perform comparably (.95 vs .94). This mirrors the HYBRID trade-off: process-enriched training slightly reduces discriminative power on finished texts.

Aggregate performance. Across all time thresholds, COMBINED achieves the highest QWK in every condition (.67 *NoChat*, .82 *Chat*, .79 combined), outperforming HYBRID (.62/.73/.73) and FULL (.58/.57/.62). This is most relevant for practical applications where a single model serves writers at any stage of composition.

Condition asymmetry. The advantage of COMBINED over FULL is consistently larger in the *Chat* condition (+.26 vs. +.09 in aggregate). This likely reflects both the richer keystroke features available in the *Chat* condition (e.g., interface switches, time allocation across areas) and the right-skewed score distribution, which may amplify the benefit of supplementary process information for distinguishing among high-scoring essays.

6 Conclusion

We investigated whether keystroke logging data can contribute to automated essay scoring beyond its established role in describing writing processes. Additionally, we present the first scoring experiments on the PISA 2025 FLA dataset, using nearly 4,000 EFL writing sessions to compare process-based, text-based, and combined approaches across writing time thresholds, demonstrating its utility for writing process research.

Our results show that process features carry genuine predictive signal for essay quality. At early writing stages, when texts are too short for conventional scoring, process features distinguish writers at performance levels that text-based models cannot yet reach. At 5 minutes, process-feature models achieve QWK scores of .32 (*NoChat*) and .59 (*Chat*), substantially outperforming DeBERTa trained on full texts (.15 and .21). This advantage diminishes as essays grow longer: at completion, text-based scoring clearly dominates (.86/.95 vs. .76/.91), showing that process features complement rather than replace content-based assessment.

The combined text-and-process model offers the best trade-off across the writing timeline. In the aggregate evaluation across all time thresholds, it outperforms both the text-only and hybrid models, achieving the highest QWK in every condition. This extends reliable prediction: in *Chat*, the combined model reaches QWK above .70 at 10 minutes, versus 20 minutes for the text-only model.

Across all models and time thresholds, prediction quality was consistently higher in the *Chat* condition, likely because chat-assisted students produced more structured, content-rich text earlier in the session, improving scoring at intermediate stages. In addition, the right-skewed score distribution in this condition may also contribute to the observed differences.

While our results demonstrate that process features improve predictive accuracy at intermediate

writing stages, whether these predictions translate into effective formative feedback remains an open question for future intervention studies. Further work should investigate which process features drive prediction at different writing stages and evaluate the approach on human-annotated training data.

Limitations

Our models are trained on silver-standard labels from a linear regression model rather than human annotations, although all evaluations are against human-annotated gold scores. The consistent performance advantage in the *Chat* condition may partly reflect its right-skewed score distribution rather than richer behavioural signal alone. Our dataset comprises German high school students writing argumentative EFL essays in a standardised assessment context; generalisability to other populations, tasks, and settings remains to be established. Finally, we demonstrate predictive accuracy at intermediate time points, not the pedagogical effectiveness of delivering such predictions as formative feedback in practice.

Acknowledgments

We used large language model assistants for code development support and language editing. All experimental design, analysis, and scientific interpretation are our own.

References

- Laura K. Allen, Matthew E. Jacovina, Mihai Dascalu, Rod D. Roscoe, Kevin M. Kent, Aaron D. Likens, and Danielle S. McNamara. 2016. {ENTER}ing the time series {space}: Uncovering the writing process through keystroke analyses. In *Proceedings of the 9th International Conference on Educational Data Mining 2016*, pages 22–29.
- Rianne Conijn, Christine Cook, Menno van Zaanen, and Luuk Van Waes. 2022. [Early prediction of writing quality using keystroke logging](#). *International Journal of Artificial Intelligence in Education*, 32(4):835–866.
- Elena Cotos. 2015. [Automated writing analysis for writing pedagogy: From healthy tension to tangible prospects](#). *Writing and Pedagogy*, 7(2-3):263–281.
- Linda Flower and John R. Hayes. 1981. [A cognitive process theory of writing](#). *College Composition and Communication*, 32(4):365–387.

- Steve Graham, Michael Hebert, and Karen R. Harris. 2015. [Formative assessment and writing: A meta-analysis](#). *The Elementary School Journal*, 115(4):523–547.
- John Hattie and Helen Timperley. 2007. [The power of feedback](#). *Review of Educational Research*, 77(1):81–112.
- John R. Hayes. 2012. [Modeling and remodeling writing](#). *Written Communication*, 29(3):369–388.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *CoRR*, abs/2111.09543.
- Stefan D. Keller, Johanna Fleckenstein, Maleika Krüger, Olaf Köller, and André A. Rupp. 2020. [English writing skills of students in upper secondary education: Results from an empirical study in switzerland and germany](#). *Journal of Second Language Writing*, 48:100700.
- Stefan D. Keller, Julian Lohmann, Ruth Trüb, Johanna Fleckenstein, Jennifer Meyer, Thorben Jansen, and Jens Möller. 2024. [Language quality, content, structure: What analytic ratings tell us about efl writing skills at upper secondary school level in germany and switzerland](#). *Journal of Second Language Writing*, 65:101129.
- Learning Agency Lab. 2024. [Automated essay scoring 2.0](#). Kaggle Competition. 1st-place writeup: <https://www.kaggle.com/competitions/learning-agency-lab-automated-essay-scoring-2/discussion/516571>.
- Mariëlle Leijten and Luuk Van Waes. 2013. [Keystroke logging in writing research](#). *Written Communication*, 30:358–392.
- Shengjie Li and Vincent Ng. 2024. [Automated essay scoring: A reflection on the state of the art](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.
- Julian F. Lohmann, Fynn Junge, Jens Möller, Johanna Fleckenstein, Ruth Trüb, Stefan Keller, Thorben Jansen, and Andrea Horbach. 2025. [Neural networks or linguistic features? Comparing different machine-learning approaches for automated assessment of text quality traits among L1- and L2-Learners’ argumentative essays](#). *International Journal of Artificial Intelligence in Education*, 35:1178–1217.
- Nils-Jonathan Schaller, Yuning Ding, Thorben Jansen, and Andrea Horbach. 2025. [Don’t score too early! evaluating argument mining models on incomplete essays](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 345–355, Vienna, Austria. Association for Computational Linguistics.
- Valerie J. Shute. 2008. [Focus on formative feedback](#). *Review of Educational Research*, 78(1):153–189.
- Hansel Susanto, Alexander Agung Santoso Gunawan, and Muhammad Fikri Hasani. 2024. [Development of automated essay scoring system using DeBERTa as a transformer-based language model](#). In *Data Analytics in System Engineering. CoMeSySo 2023*, volume 935 of *Lecture Notes in Networks and Systems*, pages 202–215, Cham. Springer.
- Yu Tian, Scott Crossley, and Luuk Van Waes. 2025. [The klicke corpus: Keystroke logging in compositions for knowledge evaluation](#). *Journal of Writing Research*, 17:23–60.
- Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. [Neural automated essay scoring incorporating hand-crafted features](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Nina Vandermeulen, Mariëlle Leijten, and Luuk Van Waes. 2020. [Reporting writing process feedback in the classroom. using keystroke logging data to reflect on writing processes](#). *Journal of Writing Research*, 12:109–140.
- Georgios Velentzas, Andrew Caines, Rita Borgo, Erin Pacquetet, Clive Hamilton, Taylor Arnold, Diane Nicholls, Paula Buttery, Thomas Gaillat, Nicolas Ballier, and Helen Yannakoudakis. 2024. [Logging keystrokes in writing by English learners](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10725–10746, Torino, Italia. ELRA and ICCL.

A Appendix

	w/o Chat			w/ Chat		
	All	Len.	NoLen.	All	Len.	NoLen.
5 min	.25	.05	.25	.52	.25	.52
10 min	.42	.26	.38	.52	.42	.48
15 min	.55	.42	.48	.64	.57	.55
20 min	.57	.56	.50	.71	.62	.60
25 min	.70	.66	.65	.77	.74	.58
Full	.72	.68	.67	.80	.77	.63

Table 6: QWK for linear regression baselines at each time threshold. ALL = all keystroke features; LEN = text length only; NOLEN = all features except length.

Category	Feature	Definition
Temporal	write_time	First to last keystroke (s)
	init_pause	Task start to first key (ms)
Pauses	brk_count	Pauses ≥ 2000 ms
	brk_total	Sum of pause durations (s)
	brk_mean_dur	Mean pause duration (ms)
	brk_ratio	Break / total time (0–1)
Bursts	bst_count	Production sequences
	bst_mean_len	Mean chars per burst
	bst_mean_dur	Mean burst duration (ms)
Deletions	del_count	Delete/backspace events
	del_ratio	Deletions / total keys (0–1)
	del_chars	Total chars removed
Production	key_count	Total key-down events
	txt_len_chr	Char count at threshold
	chars_per_min	Chars / active time (cpm)
	pp_ratio	Process-to-product ratio
Navigation	nav_count	Cursor reposition events
	cp_count	Paste events
	lin_index	Forward cursor transitions (0–1)
	tgt_switches	Interface area switches
Chat [†]	t_editor	Time in text editor (s)
	t_task	Time on task description (s)
	t_chat	Time in chat window (s)
	t_prompt	Time in chat prompt (s)
	t_none	Time in no interface area (s)

Table 7: Keystroke features. [†] Chat only; set to zero for NoChat.

Table 8: Pearson correlations of keystroke features with text length (r_{len}) and holistic score (r_{score}) for the NoChat and Chat conditions (all tasks combined, $N = 3,882$). Features are sorted by $|r_{score}|$ in the NoChat condition. [†] denotes the LEN baseline feature.

Feature	NoChat		Chat	
	r_{len}	r_{score}	r_{len}	r_{score}
total_keystrokes	.44	.37	-.02	-.01
break_count	.29	.34	-.17	-.14
burst_count	.25	.28	-.21	-.20
total_writing_time	.26	.28	-.02	.03
process_product_ratio	-.24	-.27	-.24	-.32
linearity_index	.22	.23	-.07	-.08
burst_mean_length_char	.22	.20	.11	.11
initial_pause	.05	-.19	-.03	-.16
break_ratio	-.24	-.18	.10	.06
break_mean_duration	.01	-.15	.09	.14
deletion_ratio	.02	-.08	.02	-.02
burst_mean_duration	.03	-.04	.02	.04
txt_len_chr [†]	1.00	.73	1.00	.73
chars_per_minute	.56	.41	.63	.46

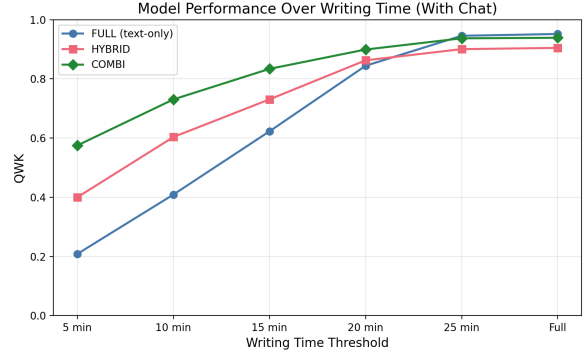


Figure 3: Performance of all 3 DeBERTa models only on Chat condition.

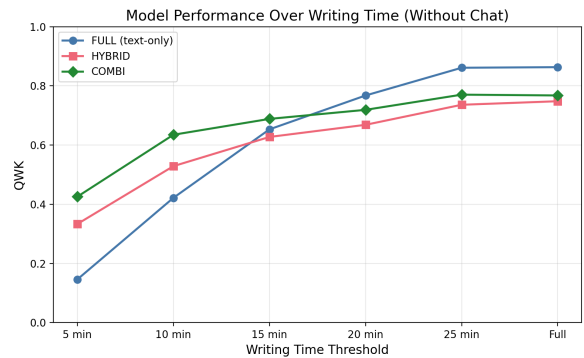


Figure 4: Performance of all 3 DeBERTa models only on NoChat condition.

EduMUSE: A Multimodal Educational Dataset with Automatically Extracted Instructional Context

Andreea Dutulescu and Stefan Ruseti and Mihai Dascalu

National University of Science and Technology POLITEHNICA Bucharest
{andreea.dutulescu, stefan.ruseti, mihai.dascalu}@upb.ro

Danielle S. McNamara

Arizona State University
danielle.mcnamara@asu.edu

Abstract

Research in AI applied to education increasingly relies on large-scale, high-quality datasets to support the development and evaluation of learning analytics and intelligent educational systems. Open educational resources provide a promising foundation, yet few datasets integrate structured instructional content with assessment materials in a multimodal form. In this study, we introduce a large-scale multimodal educational dataset (EduMUSE - Educational Multimodal Understanding & Solution Dataset) constructed from OpenStax undergraduate textbooks across multiple domains. The dataset integrates hierarchically structured instructional text, figures, exercises, and, when available, official solutions. For exercises with solutions, we introduce an automatic method that associates each exercise with a focused instructional subsection rather than entire textbook chapters, estimating subsection relevance via solution likelihood under candidate contexts using a vision-language model. We analyze the impact of contextualization on the behavior of vision-language models across different contexts. Results indicate that subsection-level instructional context has a measurable impact on model performance, with variation across model scales and task formulations. The dataset and code are released as open source at <https://github.com/upb-nlp/BEA-EduMUSE/> to support reproducible research in multimodal educational modeling and to facilitate generating similar datasets using our approach.

1 Introduction

The effectiveness of educational technologies largely depends on the quality and structure of the datasets used for training and evaluation (Chen et al., 2025). This dependency has become more evident as the use of Large Language Models (LLMs) in educational applications, such as question answering, question generation, tutoring systems, and

automated assessment, has increased (Wang et al., 2024; Dong et al., 2024; García-Méndez et al., 2025). LLMs typically require large amounts of data and are sensitive to noise, inconsistencies, and misalignment with instructional goals. As a result, the availability of clean, well-organized, and pedagogically relevant datasets remains an important factor in NLP research for education.

Many existing datasets for educational question answering and question generation are created using crowdsourcing platforms. While this approach facilitates scalable data collection, it often involves contributors without formal training in education or domain-specific expertise (Díaz et al., 2022; Klie et al., 2024). Moreover, the availability of datasets targeting college-level education is more limited, where materials often involve formal reasoning, domain-specific terminology, and structured explanations, which increases the difficulty of dataset construction and annotation.

In addition to textual content, many academic disciplines rely extensively on visual elements such as diagrams, figures, graphs, and illustrations (Jewitt, 2012). These elements play a central role in instruction and assessment across domains such as mathematics, physics, biology, chemistry, and engineering. However, a large portion of existing educational datasets remains text-centric, which constrains research on multimodal models that integrate visual and textual information in learning-oriented tasks.

To help address these gaps, we construct a large-scale multimodal educational dataset based on college-level textbooks from OpenStax (OpenStax, 2026). The dataset includes textbook text, images, exercises, and corresponding solutions across multiple domains. By sourcing content from openly licensed instructional materials, the dataset remains aligned with established curricula while enabling its reuse for research purposes. The inclusion of images supports the study of multimodal educational

tasks. An additional challenge in educational question answering is identifying the relevant instructional context required to solve a given exercise. In realistic learning scenarios, learners are expected to rely on specific portions of the instructional material rather than the full textbook. To model this aspect, we introduce an automatic, LLM-based method for extracting instructional context. The method evaluates candidate text segments based on the likelihood that an LLM produces a correct answer.

The main contributions of this work are as follows:

- The construction of EduMUSE, a large-scale, college-level multimodal educational dataset derived from OpenStax textbooks, including instructional text, images, exercises, and reference solutions across multiple domains
- An automatic, likelihood-based method for associating exercises with fine-grained instructional subsections, enabling compact contextual grounding for educational question answering
- An empirical analysis of vision–language model performance that quantifies the impact of instructional context, visual information, and model scale on solution likelihood and multiple-choice accuracy

2 Related Work

Large-scale question-answering datasets have played a central role in advancing machine reading comprehension and, more recently, LLMs. Early corpora primarily focused on factoid question answering over encyclopedic text, whereas more recent efforts have explored multi-hop reasoning, educational assessment, and synthetic data generation. However, existing datasets often trade off between scale, pedagogical structure, reasoning depth, and the explicit inclusion of instructional context and solutions.

Among the earliest and most influential datasets is SQuAD by [Rajpurkar et al. \(2016\)](#), which introduced large-scale span-based question answering over Wikipedia articles. Questions were designed by crowdworkers. Given its size and simplicity, SQuAD became a de facto benchmark for reading comprehension models. However, its questions largely emphasize local information retrieval and lexical matching, with limited requirements

for complex reasoning or synthesis across sources. To address these limitations, [Yang et al. \(2018\)](#) introduced HotpotQA, which extended the single-document paradigm by introducing multi-hop question answering over Wikipedia articles. Questions require aggregating evidence across passages, and the dataset explicitly annotates supporting facts, enabling more fine-grained evaluation of reasoning. While HotpotQA represents a significant step toward compositional reasoning, it remains grounded in encyclopedic knowledge and inherits the limitations of crowd-authored questions, with less attention paid to pedagogical intent and question quality. Moreover, both SQuAD and HotpotQA are now relatively dated and were designed prior to the emergence of instruction-following LLMs, limiting their suitability for evaluating modern educational reasoning capabilities.

In contrast to crowdsourced encyclopedic datasets, [Xu et al. \(2022\)](#) adopted an explicitly educational perspective for FairytaleQA. The dataset is curated by educational experts and is based on children’s storybooks, with questions designed to assess narrative understanding and basic comprehension skills appropriate for young learners. While FairytaleQA benefits from expert-driven design and pedagogical alignment, its scope is intentionally narrow: questions are simple, targeted at early education, and do not capture advanced academic domains.

A broader educational focus is found in EduQG by [Hadifar et al. \(2023\)](#), who introduced a multiple-choice question dataset derived from OpenStax textbooks. Questions are annotated with Bloom’s taxonomy levels ([Bloom et al., 1956](#)), enabling analysis of cognitive complexity across formats. EduQG represents an important step toward structured educational QA, particularly through its explicit linkage to pedagogical theory. However, the dataset focuses exclusively on multiple-choice questions, limiting its applicability for tasks involving explanatory reasoning or open-ended answer generation.

In a recent benchmark, [Dinh et al. \(2024\)](#) targeted higher education and expert-level assessment. SCiEx targets university exam questions and covers domains such as Artificial Intelligence, Natural Language Processing, Human–Computer Interaction, and Databases. Despite the difficulty and topical diversity of the questions, the dataset notably lacks accompanying contextual material, requiring models to rely almost entirely on parametric knowl-

edge rather than contextualized reasoning.

Similarly, [Singh et al. \(2024\)](#) propose SCIDQA, a deep reading comprehension dataset derived from scientific papers, with questions originating from peer-review discussions and answers from author responses. While positioned as a dataset for deep scientific understanding, the alignment between questions and the provided paper content is not always explicit. In many cases, answering a question may require background knowledge or interpretative reasoning beyond what is directly stated in the text, raising questions about the dataset’s suitability for controlled reading comprehension evaluation. As reasoning becomes more complex, [Zhu et al. \(2024\)](#) introduced FanOutQA, a multi-hop, multi-document QA benchmark in which answering a question requires aggregating information from a large number of articles. The dataset explicitly decomposes each question into intermediate sub-questions and associated documents, offering transparency into the reasoning process. Evaluation across several LLMs reveals substantial performance gaps, highlighting the difficulty of large-scale information aggregation. However, FanOutQA contains only development and test splits and is limited in size, making it unsuitable for model training and educational use. Additionally, the questions are primarily factual and analytic rather than pedagogically motivated.

An alternative line of work involves generating synthetic datasets. [Gong et al. \(2025\)](#) proposed PhantomWiki, a framework for on-demand dataset creation by generating a synthetic universe of entities, relations, and documents, followed by grammatically generated questions. This approach enables precise control over the complexity of reasoning and the difficulty of retrieval. However, the resulting texts and questions are highly artificial, with limited linguistic naturalness and no grounding in authentic educational content. As a result, while PhantomWiki is valuable for the evaluation of controlled reasoning, it is less suitable for modeling real-world learning scenarios. Similarly, [Naeem et al. \(2025\)](#) introduced a synthetic educational QA dataset to assess grade-level adaptability in LLMs. While it addresses an important dimension of educational alignment, the reliance on synthetic generation raises concerns regarding linguistic diversity, authenticity, and alignment with real instructional materials.

Taken together, existing educational datasets span a wide spectrum of scale, reasoning depth,

and pedagogical grounding. Large encyclopedic datasets offer scale but limited educational structure; curated educational datasets provide pedagogical alignment but are often restricted in complexity or format; expert-level and scientific benchmarks lack explicit context; and synthetic datasets enable control at the cost of ecological validity. These limitations motivate the need for a multimodal educational dataset that jointly provides rich context, well-designed questions, and explicit solutions, enabling the study of grounded reasoning, explanation generation, and learning-oriented evaluation in realistic educational settings.

3 EduMUSE Construction

This section describes the construction of the multimodal educational dataset - EduMUSE - (Educational Multimodal Understanding & Solution Dataset) - and the procedure for augmenting exercises with relevant instructional context. The method consists of two main components: large-scale scraping and normalization of college-level instructional materials from OpenStax textbooks, and an automatic context selection procedure that identifies the most relevant instructional subsections for each exercise using a likelihood-based criterion derived from a vision-language model.

3.1 Dataset Scraping

3.1.1 Source Selection

We construct EduMUSE using textbooks published by OpenStax, an initiative that provides openly licensed, college-level textbooks across multiple academic disciplines ([OpenStax, 2026](#)). OpenStax materials are aligned with standard undergraduate curricula and include a rich combination of instructional text, figures, worked examples, exercises, and solutions, making them well-suited for educational NLP tasks. EduMUSE covers textbooks from the following domains: Business, Computer Science, Humanities, Mathematics, Nursing, Science, and Social Sciences. These domains were selected to ensure diversity in reasoning styles, representational formats, and instructional practices, ranging from formal symbolic reasoning to conceptual and narrative explanations.

3.1.2 Content Extraction

OpenStax does not provide a dedicated API or bulk export mechanism for extracting structured datasets. As a result, data collection was performed

by directly scraping the textbooks in HTML representations. Although the absence of an official scraping interface increases implementation complexity, the HTML pages follow a consistent structural schema, enabling reliable content extraction. Specifically, the HTML markup includes identifiable tags and class attributes corresponding to chapters, sections, subsections, paragraphs, figures, exercises, and solutions. These structural cues were leveraged to systematically extract and hierarchically organize the textbook content.

For each textbook, we extracted the instructional content at the paragraph level, preserving the original hierarchical organization of chapters, sections, and subsections. Paragraphs were cleaned by removing extraneous HTML elements, navigation artifacts, and formatting tags, while retaining inline mathematical expressions and emphasized text. Figures and images embedded in the instructional content were also extracted. For each figure, we retained both the image itself and its associated alternative (alt) text and caption. The alt text in OpenStax textbooks typically provides a detailed description of the visual content. Retaining this serves two purposes: it supports multimodal learning scenarios for models that process images, and it provides a textual proxy for visual information when evaluating text-only models.

3.1.3 Exercise and Solution Extraction

In addition to instructional content, we extracted end-of-chapter exercises. Exercises were identified using HTML tags specific to assessment content and are associated with a corresponding chapter. Each exercise was cleaned to remove residual HTML markup, footnotes, and styling elements. Mathematical expressions originally encoded using MathJax were converted into a normalized textual representation to ensure compatibility with language models. This normalization step produces equation strings that are syntactically explicit and easily parsable by LLMs. When available, we also scraped the official solutions linked to the exercises. Exercises with solutions typically include a hyperlink to a separate solution page, which was followed and processed in the same manner as the exercise text. Solution content was cleaned using the same normalization pipeline applied to exercises. For exercises that included images or graphs in the problem statement, the visual elements and their corresponding alt text were retained, as they may be necessary to solve the exercise.

3.1.4 Limitations of Chapter-Level Context

While exercises are associated with chapters in the textbook structure, chapters are often extensive, containing thousands of tokens. Providing the full chapter as context for an exercise is impractical for many language models due to context length limitations. Even for long-context models, such large contexts can lead to diluted attention, increased inference cost, and spurious reliance on irrelevant information. This motivates the need for a more granular approach to identifying the instructional content most relevant to each exercise.

3.2 Context Augmentation

3.2.1 Motivation

In realistic educational settings, learners solve exercises by relying on specific instructional segments rather than the entire textbook chapter. However, existing datasets typically associate exercises with coarse-grained contexts (e.g., entire chapters or no context). To model a more realistic and pedagogically meaningful scenario, we aim to identify the most relevant instructional subsection for each exercise.

Each textbook chapter is hierarchically organized into sections, which are further subdivided into subsections. These subsections generally correspond to coherent instructional units focused on specific concepts, methods, or examples. As such, we consider subsections as candidate instructional contexts and aim to select, for each exercise, the subsection that provides the most relevant information for solving it.

3.2.2 Relevant Subsection Selection

The context augmentation procedure is applied to exercises for which an official solution is available. The presence of a reference solution enables an objective evaluation of how well a given instructional context supports a model’s correct answer generation.

For a given exercise with question text e , a reference solution s , and a set of candidate subsections $c_1, \dots, c_N$ from the corresponding chapter, we evaluate the relevance of each subsection by measuring how well it supports the generation of the correct solution. Formally, for each candidate subsection c_i , we compute the conditional probability: $P(s \mid c_i, e)$, where c_i and e are provided in the prompt, as instructional context and exercise statement. This probability is estimated using the

token-level logits of an instruction-tuned vision-language model. The probability is computed by decomposing the solution into a sequence of tokens $s = (s_1, \dots, s_T)$ and summing the log-probabilities assigned by the model:

$$\log P(s \mid c_i, e) = \sum_{t=1}^T \log P(s_t \mid s_{<t}, c_i, e) \quad (1)$$

where e is the target exercise, c_i is the instructional context, s is the annotated solution, and t iterates over the solution’s tokens.

This formulation captures how consistently the model predicts the annotated solution when conditioned on a given subsection.

For each exercise, we compute $P(s \mid c_i, e)$ for all candidate subsections within the chapter. The subsection that yields the highest probability is selected as the most relevant instructional context: $c^* = \arg \max_{c_i} P(s \mid c_i, e)$. This selected subsection is then associated with the exercise as its contextual grounding.

3.2.3 Experimental Setup

We employ Qwen2-VL-7B-Instruct¹ (Team et al., 2024), an instruction-tuned vision-language model that supports both textual and visual inputs. This choice allows the context evaluation procedure to incorporate not only textual instructional content but also associated images when present. Moreover, Qwen models allow for multiple images in the input, unlike other VLM families. For exercises or subsections that include figures, the images are provided according to the model’s input specifications and are correctly positioned in their original locations within the text.

3.2.4 Outcome

After applying the previous procedure, each exercise with an available solution in EduMUSE becomes paired with a specific instructional subsection that is empirically validated to be maximally supportive of correct answer generation by a vision-language model. This context augmentation enables more realistic evaluation of educational question answering and reasoning tasks while reducing context length and mitigating the effects of irrelevant instructional material.

¹<https://huggingface.co/Qwen/Qwen2-VL-7B-Instruct>

3.3 Dataset Characterization

This subsection provides a quantitative overview of EduMUSE after scraping, normalization, and context augmentation. EduMUSE was constructed from 39 textbooks across multiple academic domains. From these sources, we extracted 40,971 exercises in total. Among these, 7,195 exercises (17%) are multiple-choice questions (MCQs), reflecting the diversity of assessment formats present in the original textbooks.

A subset of the exercises includes official reference solutions. In total, 15,297 exercises (37%) are paired with solutions, of which 2,037 are MCQs. These exercises constitute the primary subset used for context augmentation, as a reference solution is required to compute solution likelihoods during subsection selection.

EduMUSE also contains substantial visual information. A total of 2,239 exercises include one or more images directly in the problem statement, such as diagrams, plots, or illustrative figures. In addition, 8,595 exercises are associated with instructional subsections that contain images. This enables the evaluation of multimodal reasoning scenarios in which visual content appears either in the question, the instructional context, or both.

Figure 1 illustrates the distribution of the number of exercises across the textbooks included in EduMUSE. The majority of textbooks contain a substantial number of exercises with official solutions, while a non-negligible subset includes no exercises with solutions. In addition, only a limited number of textbooks provide exercises in the form of multiple-choice questions. By contrast, a significant proportion of textbooks feature multiple exercises that incorporate visual elements, such as diagrams or figures, into the problem statement or instructional context.

In terms of instructional content granularity, textbook chapters are relatively long, averaging 13,133 words and reaching a maximum of 64,511 words. By contrast, subsections are significantly shorter, with an average length of 200 words and a maximum of 5,921 words. This disparity highlights the practical importance of subsection-level context selection, as subsections provide focused instructional units that are substantially more manageable for language and vision-language models than full chapters.

Overall, EduMUSE supports a wide range of educational modeling scenarios, including text-only

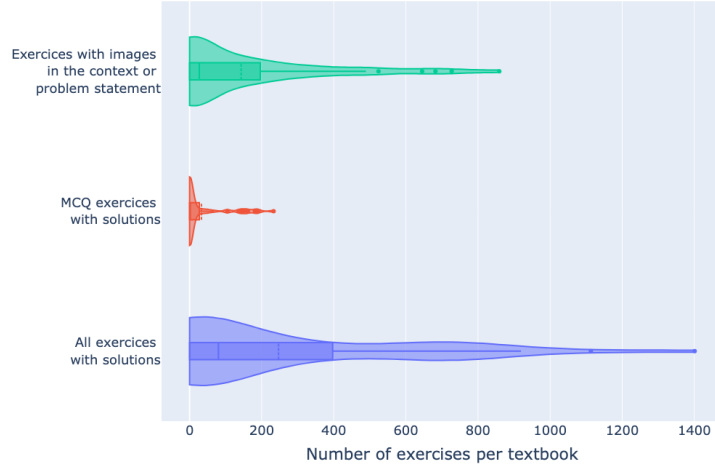


Figure 1: The distribution of the number of exercises per textbook in the OpenStax dataset.

and multimodal question answering, solution generation, and context-aware reasoning. The combination of diverse domains, varied exercise formats, explicit solution annotations, and fine-grained instructional context makes EduMUSE suitable for evaluating models under realistic instructional conditions.

4 EduMUSE Validation via VLM Probes

We use vision–language models (VLMs) on EduMUSE as probes to validate that it encodes meaningful instructional structure, with particular focus on the contributions of visual information and the selected instructional context (Section 3.2). We employ a VLM of a different family than the language model used for context augmentation while constructing EduMUSE. This separation ensures that the instructional signals encoded in the dataset generalize to newer and stronger vision–language models rather than reflecting model-specific biases. Specifically, we use Qwen3-VL-[8/32]B-Instruct² (Yang et al., 2025) as the answering model, evaluating two parameter sizes: 8B and 32B.

4.1 Probabilistic Evaluation of Annotated Solutions

In the first evaluation setup, we consider exercises for which a reference solution is explicitly annotated in the dataset. For each such exercise, we compute the probability that the model generates the correct solution, following Equation 1

²<https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct>, <https://huggingface.co/Qwen/Qwen3-VL-32B-Instruct>

described in Section 3.2. This formulation allows us to assess how different forms of context influence the model’s likelihood of producing the correct answer, without relying on discrete generation or heuristic matching.

We consider three contextual conditions: a) *full* multimodal context: the complete instructional context, including both text and images; b) *text-only* context: the full textual context, with images removed but their associated alternative text retained; and c) *no* context: the exercise prompt alone, without any instructional context. For each exercise and model size, we compute the token-level likelihood that the model assigns to the reference (annotated) solution, given the exercise prompt and a specified context. This yields three probabilities: one with the full multimodal instructional context, one with text-only context, and one with no additional context. To enable meaningful comparison across exercises with widely varying base difficulty, we compute within-exercise probability ratios between (i) the full multimodal context and the text-only context, and (ii) the full multimodal context and the no-context condition.

To summarize these ratios across EduMUSE, we report the median (50th percentile) ratio for each model and comparison. Using the median reduces sensitivity to extreme values arising from individual exercises where probabilities are unusually small or large. Therefore, the resulting values represent a *typical multiplicative increase in solution likelihood attributable to instructional and/or visual context*. Table 1 reports these median ratios for both the 8B and 32B models.

The purpose of this evaluation is not to achieve

state-of-the-art problem-solving accuracy, but to verify that EduMUSE encodes instructional structure in a form that is operationally detectable by contemporary multimodal models and that confidence substantially increases when appropriate context is provided. If the extracted instructional context is meaningful, then conditioning on it should systematically increase the likelihood assigned to expert-annotated solutions—particularly in exercises where visual material plays a pedagogical role. The observed probability gains therefore serve as evidence that EduMUSE captures instructional signals that models can exploit, rather than as a metric of absolute task performance.

The results indicate a consistent and substantial benefit from providing instructional context, particularly when visual information is included. For the larger Qwen3-32B model, the median probability of generating the correct solution is approximately $13\times$ higher when the full multimodal context is provided compared to a text-only context without images. In contrast, the smaller Qwen3-8B model exhibits a much stronger dependency on visual information, with a $366\times$ increase in median probability in the same comparison. A similar trend is observed when comparing the full multimodal context to the no-context condition, with probability ratios of $94\times$ for the 32B model and $462\times$ for the 8B model.

These results suggest that larger models are more robust to reductions in available instructional content. The Qwen3-32B model likely compensates for missing contextual information by leveraging external knowledge encoded in its parameters, resulting in smaller performance degradation when images or context are removed. Conversely, the smaller Qwen3-8B model relies more heavily on explicit instructional signals, particularly multimodal cues, to produce correct solutions. Nevertheless, across both model scales, the inclusion of multimodal instructional content consistently yields higher probabilities of correct answers. This highlights the importance of multimodal context for educational question answering across diverse exercise types, including structured problems and open-ended responses.

4.2 Multiple-Choice Question Answering Performance

In the second evaluation setup, we assess model performance on multiple-choice questions (MCQs). For each MCQ, models are prompted to select the

correct answer option, constrained to output only a single letter corresponding to one of the choices. To ensure consistent and unambiguous outputs, we employ structured decoding that restricts generation to the available answer set.

As in the probabilistic setup, we consider three contextual conditions: full multimodal context, text-only context (without images but including alternative text), and no context. Performance is measured by accuracy, defined as the percentage of exercises answered correctly. In this setting, multiple-choice answering is used as a stress test for the dataset’s contextual structure under constrained output conditions. Unlike probabilistic solution likelihood, this formulation requires models to compress potentially rich instructional information into a single discrete decision, making it particularly sensitive to context length and model capacity.

Table 2 reports accuracy results across all MCQ exercises in the dataset, while Table 3 considers only exercises whose instructional context contains at least one image. This stratification enables a more direct analysis of the impact of visual information on question answering performance. Comparisons across model sizes and context conditions reveal how scaling and multimodal inputs jointly affect VLMs’ capability to solve college-level textbook problems.

In contrast to the probabilistic evaluation presented earlier, this setup requires explicit answer generation under a constrained output format. Under these conditions, we observe that the smaller Qwen3-8B model is negatively affected by the inclusion of large instructional contexts. For both the full set of exercises and the subset containing images, the 8B model achieves the highest accuracy in the no-context setting, with performance decreasing as more context is provided. This suggests that, for smaller models, long or information-dense contexts may negatively impact the generation. Nevertheless, this behavior does not impact subsection selection, as the selection is performed based on likelihood derived from model logits rather than on the generated output itself.

This behavior is consistent with findings reported by Hsieh et al. (2024) who show that smaller models experience more pronounced performance degradation as context length increases. Limited parameter capacity and narrower effective attention spans reduce their capability to attend to relevant information in long contexts. As a result, additional

Context Comparison	Qwen3-8B	Qwen3-32B
Full vs. Text-only Context	366	13
Full vs. No Context	462	94

Table 1: Median probability ratios (50th percentile) comparing full multimodal context against reduced-context settings for Qwen3 models.

Context Condition	Qwen3-8B	Qwen3-32B
Full Multimodal Context	31.66%	41.04%
No Context	37.65%	40.89%

Table 2: Multiple-choice question accuracy for Qwen3 models under different instructional context conditions across all exercises.

Context Condition	Qwen3-8B	Qwen3-32B
Full Multimodal Context	45.05%	58.97%
Text-only Context	47.39%	58.89%
No Context	53.74%	58.81%

Table 3: Multiple-choice question accuracy for Qwen3 models on exercises whose instructional context contains at least one image.

instructional content, while potentially useful, can hinder decision-making in constrained generation tasks such as multiple-choice answering.

In contrast, the Qwen3-32B model exhibits stable or slightly improved performance when an instructional context is provided. For the full exercise set, accuracy is comparable across the whole-context and no-context conditions, whereas for exercises containing images, the full multimodal context yields slightly higher accuracy. This indicates that larger models are better able to leverage extended and multimodal contexts, likely due to increased parameter capacity and more expressive attention mechanisms that enable effective filtering of relevant information.

Finally, across all evaluation conditions, we observe a clear performance gain when moving from the 8B to the 32B model, confirming the expected scaling effects of model size on educational question answering accuracy.

5 Conclusions and Future Directions

This work introduces EduMUSE, a large-scale, multimodal educational dataset derived from college-level OpenStax textbooks and designed to support research on solution generation and context-aware question answering in realistic instructional settings. By systematically extracting textbook text, images, exercises, and official solutions across multiple academic domains, the dataset

addresses several limitations of existing educational benchmarks, including the lack of pedagogical grounding, insufficient multimodal content, and coarse or absent instructional context.

A key contribution of this work is the proposed automatic context augmentation method, which associates each exercise with a focused instructional subsection selected based on the solution likelihood predicted by a vision–language model. This approach produces compact contexts that better reflect how learners engage with educational materials. The resulting subsection-level grounding mitigates the challenges posed by long chapters while preserving pedagogical coherence. By validating the dataset through controlled model probes, this work aligns with goals of educational technology: understanding how structure, context, and representation shape performance in learning-oriented tasks.

Overall, the dataset and accompanying analyses provide a foundation for studying multimodal educational reasoning under controlled yet realistic conditions. The combination of authentic instructional content, explicit solutions, fine-grained context, and diverse domains enables future work on curriculum-aware modeling, explanation generation, context selection, and the interaction between visual and textual information in learning-oriented tasks. The dataset also offers opportunities for investigating the impact of context length and model scaling behavior.

Limitations

Despite the strengths of EduMUSE, several limitations should be acknowledged.

First, although the dataset spans multiple academic domains, it is constructed from a finite set of OpenStax textbooks that primarily cover widely taught undergraduate subjects. As a result, the dataset may not fully capture the breadth of specialized, advanced, or less commonly taught topics. Extending the dataset to include a broader range of disciplines, including graduate-level materials and niche subject areas, would improve its coverage and applicability.

Second, the proposed context selection method relies on model-derived likelihoods to identify the most relevant instructional subsection for each exercise. While this approach provides a principled and scalable mechanism for automatic context assignment, it is inherently dependent on the behavior and calibration of the underlying model. In particular, although likelihood-based scoring using logits offers a reasonable proxy for contextual relevance, it does not disentangle whether improvements arise from genuinely informative context or from increased model confidence in generating specific token sequences.

Overall, these limitations highlight important directions for future work. Addressing these aspects should aim to improve the reliability and generalizability of multimodal educational datasets and understanding on how models leverage instructional context in realistic learning scenarios.

Acknowledgments

The research reported here was supported by the project “Romanian Hub for Artificial Intelligence - HRIA”, Smart Growth, Digitization and Financial Instruments Program, 2021–2027, MySMIS no. 351416, by the Institute of Education Sciences, U.S. Department of Education, through Grant R305T240035 to Arizona State University, and the grant of the Academy of Romanian Scientists, AOSR-TEAMS-IV Edition 2025-2026 “Digital Transformation in Science”. The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

References

- B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain*. David McKay, New York.
- Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. 2025. Revisiting scaling laws for language models: The role of data quality and training strategies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23881–23899.
- Mark Díaz, Ian Kivlichan, Rachel Rosen, Dylan Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Remi Denton. 2022. Crowdsheets: Accounting for individual and collective identities underlying crowdsourced dataset annotation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2342–2351.
- Tu Anh Dinh, Carlos Mulloy, Leonard Bärmann, Zhaolin Li, Danni Liu, Simon Reiß, Jueun Lee, Nathan Lerzer, Jianfeng Gao, Fabian Peller-Konrad, and 1 others. 2024. Sciex: Benchmarking large language models on scientific exams with human expert grading and automatic grading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11592–11610.
- Bingyu Dong, Jie Bai, Tao Xu, and Yun Zhou. 2024. Large language models in education: A systematic review. In *2024 6th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 131–134. IEEE.
- Silvia García-Méndez, Francisco de Arriba-Pérez, and María del Carmen Somoza-López. 2025. A review on the use of large language models as virtual tutors. *Science & Education*, 34(2):877–892.
- Albert Gong, Kamilė Stankevičiūtė, Chao Wan, Anmol Kabra, Raphael Thesmar, Johann Lee, Julius Klenke, Carla P Gomes, and Kilian Q Weinberger. 2025. Phantomwiki: On-demand datasets for reasoning and retrieval evaluation. In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*.
- 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.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Krizan, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What’s the real context size of your long-context language models? *CoRR*.
- Carey Jewitt. 2012. *Technology, literacy, learning: A multimodal approach*. Routledge.

- Jan-Christoph Klie, Richard Eckart De Castilho, and Iryna Gurevych. 2024. Analyzing dataset annotation quality management in the wild. *Computational Linguistics*, 50(3):817–866.
- Numaan Naeem, Abdellah El Mekki, and Muhammad Abdul-Mageed. 2025. Eduadapt: A question answer benchmark dataset for evaluating grade-level adaptability in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34224–34251.
- OpenStax. 2026. Openstax: Free and open college textbooks. <https://openstax.org/>.
- 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.
- Shruti Singh, Nandan Sarkar, and Arman Cohan. 2024. Scidqa: A deep reading comprehension dataset over scientific papers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20908–20923.
- Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).
- Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, and 1 others. 2022. Fantastic questions and where to find them: Fairytaleqa—an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.
- Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37.

Opportunities and Challenges of LLMs in Education: An NLP Perspective

Sowmya Vajjala¹, Bashar Alhafni², Stefano Bannò³,
Kaushal Kumar Maurya², Ekaterina Kochmar^{2,4}

¹National Research Council, Canada, ²MBZUAI, ³University of Cambridge, ⁴UVic
sowmya.vajjala@nrc-cnrc.gc.ca, bashar.alhafni@mbzuai.ac.ae,
sb2549@cam.ac.uk, {kaushal.maurya, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

Interest in the role of large language models (LLMs) in education is increasing, considering the new opportunities they offer for teaching, learning, and assessment. In this paper, we examine the impact of LLMs on educational NLP in the context of two main application scenarios: *assistance* and *assessment*, grounding them along the four dimensions – reading, writing, speaking, and tutoring. We then present the new directions enabled by LLMs, and the key challenges to address. We envision that this holistic overview would be useful for NLP researchers and practitioners interested in exploring the role of LLMs in developing language-focused and NLP-enabled educational applications of the future.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across various tasks within and beyond NLP. The rapid adoption of LLMs and generative AI by EdTech companies such as Duolingo (Naismith et al., 2023a) and Grammarly (Raheja et al., 2023, 2024) and the development of fine-tuned models for educational use cases such as LearnLM (Team et al., 2024) are some examples of real-world impact in Education domain. The NLP community has a long history in this area, especially on problems such as automated essay scoring, grammatical error correction, and text simplification, to name a few. Naturally, there is a huge interest in using LLMs for educational applications within the community. While LLMs have undoubtedly caused a paradigm shift in this area, enabling new opportunities in writing assistance, personalization, and interactive teaching and learning, among other tasks, they also present novel challenges. In this paper, we delve into the opportunities and challenges presented by LLMs for educational applications by considering the use cases involving language, and instruction in natural

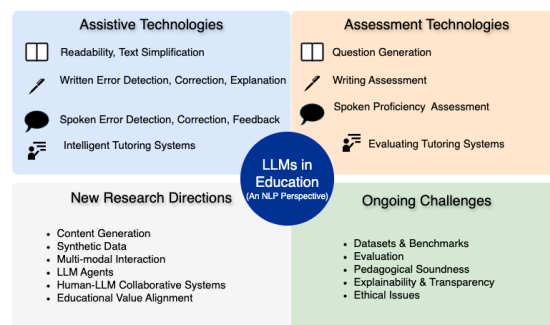


Figure 1: Overview of the paper.

language, and connect the recent developments to past NLP research in this area, outlining the path ahead.

Figure 1 shows an overview of the paper organization. After situating this paper in the context of others on the topic (§2), we discuss the state of the art by grouping it into two main topics: *assistive technologies* – meant to support students and teachers (§3) – and *assessment technologies* – meant to assess the performance of students (§4). Within each, we discuss the role of NLP and LLMs across specific aspects of education – reading, writing, speaking, and general tutoring. We then turn to some of the new directions enabled by LLMs in educational NLP (§5), point to some ongoing challenges (§6), and summarize our key insights (§7). In terms of scope, as we focus on language use, we exclude topics such as learning analytics, development of student models, measuring long-term educational outcomes, interactive classroom technologies, user studies, and similar.

2 Related Work

Post ChatGPT, there has been a lot of interest in the role of LLMs, and more generally, generative AI, in education. Several surveys and opinion pieces were published in the past two years outlining the potential use cases, ethical implications, and chal-

allenges of using LLMs in education, primarily by research groups focusing on the use of technology in education (Dempere et al., 2023; Fuchs, 2023; Kasneci et al., 2023; Yan et al., 2024). However, none of them take a closer look at methods (LLM-based or not) used to address the different use cases of LLMs in education, which is of primary interest with NLP.

NLP research has a long history of working on problems of relevance to the education domain, such as automated language assessment and grammatical error correction, among others. Burstein (2009) and Meurers (2012) give an overview of how NLP plays a role in the development of computational tools for language learning and assessment, and more recently, Vajjala (2024) has explored the role of generative AI in language learning technology. But these papers are primarily directed towards non-NLP audience and we are not aware of any such discussion directed towards an NLP audience, connecting recent developments supported by generative AI and large language models with past NLP research on these topics.

Caines et al. (2023) discuss the role of LLMs specifically in language teaching and assessment technologies from an NLP standpoint. However, they specifically focus on writing assessment, and do not address speaking assessment. Further, the use of language in the educational applications space is not limited to language learning alone, and it encompasses other areas such as content learning, as well as assessment and learning support for other subjects. Similarly, Chu et al. (2025c) recently focused on the use of LLM agents for educational applications in detail, while we specifically look at the development of language-based educational applications. Thus, compared to this existing work, we aim to give a broader perspective on the relevance of LLMs for the development of language-based educational applications, and how other NLP methods fit into the current scenarios. Where relevant, we also summarize the perspectives on the use of NLP and LLMs in this area from other related disciplines.

3 Assistive Technologies

We refer to the NLP problems focused on supporting learners and/or instructors as *assistive technologies*, and discuss them by splitting them into four groups: writing, speaking, reading, and general tutoring. Note that we focus on the recent devel-

opments and refer to the relevant surveys for the pre-LLM research, where needed.

3.1 Writing

Assistive technologies for writing primarily focus on **Grammatical Error Detection (GED)** and **Correction (GEC)**, both of which have a long-standing pedagogical value in writing assistance tool development. GEC has witnessed significant progress over the past two decades through the organization of several shared tasks (Ng et al., 2014; Bryant et al., 2019; Masciolini et al., 2025, *inter alia*). For a comprehensive overview of the GEC literature, see the survey by Bryant et al. (2023). While GEC has received much of the attention, GED has also evolved as a stand-alone task (Tetreault and Chodorow, 2008; Leacock et al., 2014; Rei and Yannakoudakis, 2016, *inter alia*).

Several recent studies have applied LLMs to (mainly English) GEC, comparing prompting methods along two dimensions: strategy (e.g., zero-shot, few-shot, chain-of-thought) and design (e.g., fluency-oriented vs. minimal edits). So far, few-shot prompting tends to outperform zero-shot, while chain-of-thought shows no clear benefit (Fang et al., 2023; Coyne et al., 2023; Wu et al., 2023a; Loem et al., 2023; Davis et al., 2024; Katin-skaia and Yangarber, 2024; Omelianchuk et al., 2024). In terms of performance, LLMs often outperform state-of-the-art models on some benchmarks such as JFLEG (Napoles et al., 2017) due to their strength in generating fluent rewrites, but underperform on larger benchmarks like CoNLL-2014 (Ng et al., 2014) and BEA-2019 (Bryant et al., 2019), which prioritize precision and minimal edits. This reflects the difficulty of controlling LLMs to make minimal, targeted corrections, which is essential in educational applications where the goal is to guide learners in revising their own errors while preserving intent (Nicholls, 2003).

LLMs have also been leveraged for **Grammatical Error Explanation (GEE)**, a task that combines GED and GEC to generate natural language explanations of learner errors. Recent work has introduced methods to guide LLMs in producing such explanations using detected edits (Kaneko and Okazaki, 2024; López Cortez et al., 2024). Song et al. (2024b) evaluated LLMs on GEE in English, German, and Chinese, showing that models often struggle to identify and explain errors, though performance improves when edits are included in the prompt. There is a growing interest in GEE for

other languages as well (Ye et al., 2025; Maity and Deroy, 2025). LLMs have also been shown to be useful in providing feedback on other aspects of language assessment such as vocabulary usage (Ortiz-Zambrano et al., 2024; Bannò et al., 2025a), discourse coherence (Naismith et al., 2023b) and analytical assessment of written texts (Bannò et al., 2024b; Stahl et al., 2024; Seßler et al., 2025; Wang et al., 2025b), indicating the growing interest in this direction. Note that some previous work on feedback comment generation for writing also pursued similar goals but preceded the widespread adoption of LLMs (Nagata, 2019; Nagata et al., 2020; Hanawa et al., 2021; Nagata et al., 2021).

3.2 Speaking

As with writing, a common application for supporting learners in speech is **spoken GEC**. However, compared to written GEC which typically works with well-formed inputs where punctuation and capitalization can aid in error detection, spoken GEC presents a distinct set of challenges which significantly complicate the task of identifying and correcting grammatical errors in speech compared to written text. Traditionally, spoken GEC systems have adopted a cascaded pipeline architecture, typically consisting of an automatic speech recognition (ASR) module to transcribe audio into text, followed by a disfluency detection module to produce fluent transcriptions, and finally a GEC module to correct grammatical errors (Lu et al., 2020, 2022). While this approach has shown some effectiveness, it is often hindered by error propagation across stages, which can degrade overall system performance.

This was followed by end-to-end approaches powered by large speech foundation models such as Whisper (Radford et al., 2023), which promise to decrease the number of compounded errors (Bannò et al., 2024a). To address the problem of the scale of data needed to build such systems, Qian et al. (2025a) explored data augmentation, and Qian et al. (2025b) describe a novel reference alignment process to reduce transcription errors which outperforms a fine-tuned multimodal LLM Lu et al. (2025) for this task.

Despite these advances, generating accurate and meaningful feedback from spoken input continues to be a significant challenge. The recent release of the Speak & Improve Corpus (Knill et al., 2024), the first publicly available speech dataset annotated for grammatical errors, and its associ-

ated challenge (Qian et al., 2024) represents a major milestone and is expected to catalyze further progress and innovation in the field.

3.3 Reading

Assistive technologies for reading in NLP primarily focus on **Automatic Readability Assessment** and **Automatic Text Simplification**. Readability Assessment refers to the task of assigning a reading level to a given text based on its language difficulty, to various target readers. Interest in this topic is almost a century old among the education researchers (e.g., Vogel and Washburne, 1928) while the NLP research has an over two decade history (Kevyn, 2014; Vajjala, 2022), and different approaches from feature-based machine learning to deep learning methods have been studied. Recent adaptation of LLMs to this problem so far seems to indicate that task-specific fine-tuned models achieve better results than zero- or few-shot prompting of LLMs (Naous et al., 2024; Wang et al., 2024g; Smădu et al., 2024). However, other work demonstrates better agreement between LLM-generated reading level judgments and human evaluations (Trott and Rivière, 2024), and Rooein et al. (2024) argue for new prompt-based evaluation metrics switching from the traditional static evaluation metrics while using LLMs.

Text Simplification refers to the task of generating text in a simpler, easier to understand language, given a more complex text (typically sentence-to-sentence). It is a well-studied area of research in NLP (Alva-Manchego et al., 2020; Štajner, 2021; Chi et al., 2023; Huang and Kochmar, 2024, *inter alia*) and the advent of LLMs resulted in a natural extension into new methods for developing datasets, generating diverse simplifications and going beyond sentence level generation. While Engelmann et al. (2024) propose to use LLMs to create datasets for text simplification research, several groups showed the effectiveness of few-shot, in-context learning for generating diverse simplifications in multiple languages (Kew et al., 2023a; Nozza and Attanasio, 2023a; Scalercio et al., 2024). Human user studies show better comprehension with LLM simplified text (Guidroz et al., 2025) but also substantial variation among human judgments (Trott and Rivière, 2024). In terms of modeling, some recent approaches utilize LLMs and multi-agentic workflows to explore document-level simplification, showing promising early results (Mo and Hu, 2024; Fang et al., 2025; Qiang et al.,

2025; Alva-Manchego et al., 2025). There is also a growing interest in personalizing text simplification through preference learning (Gao et al., 2025), generating texts at multiple levels of simplification (Farajidizaji et al., 2024; Barayan et al., 2025), domain specific simplification (Zečević et al., 2024), and elaborative simplification (Hewett et al., 2024).

3.4 Tutoring

Within the domain of general knowledge acquisition and tutoring, one of the most effective NLP-enabled tools are **Intelligent Tutoring Systems (ITS)**, in particular, dialogue-based ITSs. ITSs are defined as computerized learning environments that incorporate computational models and provide feedback based on students' learning progress (Graesser et al., 2001); for dialogue-based systems, such feedback and communication with the student are empowered by NLP models. Lack of individualized tutoring has been linked to less effective learning and increased learner dissatisfaction (Brinton et al., 2014; Eom et al., 2006; Hone and El Said, 2016), particularly in large classroom settings. This has led to the development of pre-LLM ITSs (Paladines and Ramirez, 2020), including systems focused on misconception identification (Graesser et al., 1999; Rus et al., 2013), model-tracing tutors (Rickel et al., 2002; Heffernan et al., 2008), constraint-based models (Mitrovic, 2005), and Bayesian network models (Pon-Barry et al., 2004) across educational levels.

LLM-powered ITSs promise to offer more personalized, one-on-one tutoring, enabling equitable and pedagogically sound learning experiences, which have long been known to lead to substantial learning gains (Bloom, 1984). Methods such as prompting (Wang et al., 2024c), fine-tuning (Jurenka et al., 2024), Reinforcement Learning from Human Feedback (RLHF) (Team et al., 2024), and human-in-loop deployment (Team L et al., 2025) have been used in state-of-the-art LLM-based ITSs, as they help to overcome the limitations of traditional systems by enabling more adaptive, generalizable, and effective tutoring models. Although still limited, a few models design their development to stimulate curiosity and thinking processes to improve learning through data curation pipelines or modular model architectures (Jurenka et al., 2024; Vendrell and Johnston, 2026). These approaches are grounded in key learning science principles such as active learning, stimulated curiosity, adaptive learning, meta-cognition, and reasoning.

One of the key limitations for ITSs is the scope and size of current educational datasets (Macina et al., 2023b; Wang et al., 2024c; Stasaski et al., 2020a). Thus, building large-scale, publicly available educational datasets for LLM pre-training and fine-tuning should be prioritized in the near future. The focus on domain-specific models optimized for educational tasks and methods and the development of methods to assess the long-term impact of LLM-driven tutoring on learners and educators, including analysis of pedagogical effectiveness and bias, should also be considered more closely.

4 Assessment Technologies

The assessment of writing, speaking, reading, and tutoring relies on a set of overlapping principles and techniques. Although each modality has its own unique features, they are deeply intertwined, particularly in the use of textual analysis and information extraction methods. Many of these techniques, initially developed within the domain of writing assessment, have been adapted for use in speaking, reading, as well as tutoring contexts, which we summarize in this section.

4.1 Writing

The origins of **automated writing assessment (AWA)** date back to the 1960s with the introduction of Project Essay Grade (Page, 1966, 1968). Notable progress was made in the 1990s and early 2000s that saw the emergence of the commercial systems such as e-rater® (Burstein, 2002), IntelliMetric™ (Rudner et al., 2006), and the Intelligent Essay Assessor™ (Landauer et al., 2002). In later years, Deep Neural Network (DNN) approaches have led to substantial progress (Alikaniotis et al., 2016). In particular, transformer-based models have achieved performance levels that surpass even human inter-annotator agreement (Rodriguez et al., 2019). Comprehensive overviews on AWA can be found in Beigman Klebanov and Madnani (2022) and Li and Ng (2024a,b).

Recent studies looked into the usefulness of LLMs for the assessment of second language (L2) writing, obtaining promising results (Mizumoto and Eguchi, 2023; Yancey et al., 2023a). In line with Liusie et al. (2024)'s observation that LLMs tend to perform better at comparative rather than absolute assessment, Cai et al. (2025) proposed a combined ranking-and-scoring framework that outperforms standard prompt-based approaches. While

most of the writing assessment research focused on evaluating the language proficiency aspect, a substantial amount of NLP research also focused on content assessment, in the form of **short answer scoring** (Burrows et al., 2015). LLM-based research on this topic is still emerging, and recent studies so far conclude that zero-/few-shot learning with LLMs fares poorly compared to fine-tuning approaches for this task (Chamieh et al., 2024; Ferreira Mello et al., 2025).

4.2 Speaking

Research on **automated speaking assessment (ASA)** began with relatively simple tasks, such as evaluating learners' ability to read individual words or sentences (Bernstein et al., 1990; Cucchiari et al., 1997; Franco et al., 2000). A significant milestone in this field was the development of SpeechRater system, which broadened the scope of automated assessment to include both spontaneous and read speech (Xi et al., 2008). Recent years have seen significant advancements in the field through the adoption of DNN approaches (Qian et al., 2012), and end-to-end neural-based methods have outperformed traditional systems such as SpeechRater (Chen et al., 2018b). A comprehensive survey of ASA can be found in Zechner and Evanini (2019).

Pre-trained language models have contributed to further progress in ASA (Raina et al., 2020; Wang et al., 2021) and recent research has explored speech embedding representations for applications such as mispronunciation detection and diagnosis (Wu et al., 2021b; Xu et al., 2021), automatic pronunciation assessment (Kim et al., 2022), and the evaluation of proficiency across both monologic (Bannò et al., 2023a; Park and Ubale, 2023) and conversational (McKnight et al., 2023) data. The application of speech LLMs in this domain is still in its early stages. Fu et al. (2024) developed a speech LLM for L2 assessment that achieved competitive performance, albeit limited to the specific task of pronunciation scoring. With respect to holistic assessment, Ma et al. (2025) recently explored the application of Qwen2-Audio (Chu et al., 2024), for ASA in both zero-shot and fine-tuned settings. In a recent related study, Bannò et al. (2025b) demonstrated that integrating analytic proficiency descriptors with a zero-shot, text-based LLM applied to automatic transcriptions outperforms a BERT-based grader fine-tuned for the task, and achieves competitive performance compared

to fine-tuned speech LLMs. Hirschi et al. (2025) use a modular pipeline in which acoustic features are extracted using dedicated models and injected into a text LLM. For analytical ASA, Parikh et al. (2026) fine-tuned Qwen2-Audio leveraging natural language rubrics to assess prosody, accuracy, and fluency, achieving promising results.

4.3 Reading

One commonly studied problem in NLP in the area of reading assessment is the generation of reading comprehension questions. Question generation research in educational NLP and AI community in general addressed different scenarios from form-focused questions (e.g., to check grammatical knowledge) to more content-focused reading comprehension questions, using a range of methods from syntactic structures to neural language models (Kurdi et al., 2020; Perkoff et al., 2023; Uto et al., 2023; Al Faraby et al., 2024a). LLMs were used for question generation in math domain (Christ et al., 2024; Scarlatos et al., 2024a) and for personalized question generation in general (Xiao et al., 2023a; Säuberli and Clematide, 2024a). Although English is the dominant language for research on this topic, cross-lingual transfer approaches have also been explored, and Hwang et al. (2024) show that smaller fine-tuned language models can achieve comparable performance to larger language models on this task.

While the past research was restricted to a smaller datasets, the advent of LLMs resulted in approaches to benchmark construction and generation of questions at various difficulty levels (Chen et al., 2024; Scaria et al., 2024b), and towards the development of novel evaluation approaches for automatically generated questions (Moon et al., 2024; Deroy et al., 2025). Flor (2025) presents an elaborate summary of automatic question generation research from traditional rule-based methods to generative AI in a series of articles, which can serve as a good reference for those interested in further studies. Item Response Theory (IRT, Lalor et al. (2024)), which has been widely used in psychometrics, has been explored in the context of vocabulary assessment (Ehara et al., 2012; Ehara, 2018) and developing questions for general language assessment (Settles et al., 2020) in NLP research in the past. However, research combining automatic question generation with IRT is yet to emerge in the NLP domain, and this appears to be a promising area where LLMs can help.

4.4 Tutoring

Tutoring systems have long served as embedded assessment technologies, using learner interactions to evaluate understanding and guide instruction. Early systems like PLATO used rule-based feedback and simple branching logic for assessment and remediation (Woolf, 2010). Later, ITSs incorporated expert system models and student diagnostic models to reason about domain knowledge and identify misconceptions (Clancey, 1987). By the 1990s, cognitive tutors like the Algebra Tutor employed cognitive models combined with model tracing and knowledge tracing to perform fine-grained, real-time skill assessment (Anderson et al., 1995). Other ITS approaches such as AutoTutor utilize NLP models and dialogue-based reasoning to assess deeper conceptual understanding (Nye et al., 2014).

Recent advances in LLMs have transformed tutoring systems into flexible, multi-modal assessment environments. LLM-based platforms like Khanmigo (Shetye, 2024) and Google’s LearnLM (Jurenka et al., 2024; Team et al., 2024) leverage generative AI to assess learner responses in natural language, interpret comprehension across reading, writing, and speaking tasks, and adapt instruction accordingly. Unlike more traditional methods, LLMs enable open-ended, personalized feedback across diverse learning tasks, integrating instruction and assessment seamlessly (Venugopalan et al., 2025; Wang et al., 2025a).

Despite their potential, LLM-based tutoring systems often lack rigorous validation linking their assessments to learning outcomes (Macina et al., 2023c). Few studies have examined their diagnostic accuracy (Maurya et al., 2025; Naeem et al., 2026a), adaptability across diverse learners (Wang et al., 2024e), or long-term impact on knowledge retention (Kosmyna et al., 2025). Ethical concerns such as feedback bias and transparency also remain underexplored (Mvondo et al., 2023). Future research should develop standardized evaluation frameworks and investigate how LLM-driven assessments can be aligned with pedagogical goals.

Compared to assistive technologies, it appears that there are relatively fewer cases of LLM integration into assessment approaches, although it is clearly increasing. One reason could be that assessment is likely subject to higher levels of scrutiny around reliability and validity of the models, considering the potential high stakes of the outcomes.

Despite that, what we have seen so far shows how LLMs are increasingly being used in some of the common educational tasks traditionally studied by the NLP community.

5 New Directions Enabled by LLMs

In this section, we turn to previously under-explored or new use cases enabled by LLMs across the four aspects (writing, speaking, reading, tutoring), for both assistive and assessment scenarios.

Content Generation: A relatively new task, introduced with the advent of LLMs capable of fluent text generation in multiple languages, is educational content generation according to expert-defined standards (Imperial et al., 2024), for a specified grade level (Bezirhan and von Davier, 2023; Jin et al., 2025), or for creation of evaluation and scaffolding exercises for different subjects (Xiao et al., 2023a; Malik et al., 2025). One interesting question to extend this line of research further could be on-the-fly content generation through LLMs, given a topic, grade and standard specifications, as well as the target audience, across multiple languages.

Multi-modal Interaction: Although text has been the dominant form of input in educational applications of NLP so far, multi-modal LLMs are starting to be developed (Chu et al., 2025d). Curating multi-modal content for education (Chaturvedi, 2024), multi-modal question generation (Luo et al., 2024), end-to-end spoken language grammatical error correction (Bannò et al., 2024a), low-resource language learning app development (Chu et al., 2025b), supporting listening assessment (Aryadoust et al., 2024), and evaluating handwritten exams (Liu et al., 2024b) are some recent examples. Given these diverse use cases, and given that human learning can be considered multi-modal as we gain information from multiple forms of content, modeling of multi-modal interactions in human learning and multi-modal content generation can be considered challenging and useful future possibilities to study, considering large multimodal models as the foundations.

Synthetic Data Generation for Fine-tuning: Synthetic data is increasingly being used at various stages of LLM training and fine-tuning pipelines, and education domain also started to see some new use cases for synthetic data such as aiding the development of educational chatbots and tutoring sys-

tems (Wang et al., 2024a; Fateen and Mine, 2024), spoken GEC (Karanasou et al., 2025), development of benchmark datasets for educational applications (Engelmann et al., 2024; Xu et al., 2025), and using LLMs as proxies for piloting educational assessments (Säuberli et al., 2025). Considering the advantages synthetic data provides in terms of alleviating the need for labeled training data, exploring the limits and limitations of LLM-based synthetic data generation approaches for educational applications would be an important direction for the future.

LLM Agents for Education: When LLMs are combined with components such as memory, tool use, and planning to solve complex tasks, they are referred to as *LLM agents* (Chu et al., 2025c; Tran et al., 2025). In an educational context, these additional components enable real-time adaptation, access to external resources, and planning of tailored learning paths, among other capabilities. At a high level, such agents function either as pedagogical agents or domain-specific educational agents (Chu et al., 2025c).

Pedagogical agents imitate tutors to assist students or instructors in tutoring sessions and simulate students for tasks such as piloting exam questions or training tutors. Furthermore, multiple agents can operate simultaneously in multi-agent setups like in WikiHowAgent (Pei et al., 2025), CAMEL (Li et al., 2023), AutoGen (Wu et al., 2023b), and PitchQuest (Mollick et al., 2024) to develop educational prototypes or solve complex problems. These agents can further support curiosity and higher-order thinking in learners, grounded in established learning theories such as student-first learning (Schwartz and Bransford, 1998), the zone of proximal (Vygotsky, 1978) development, and productive failure (Kapur, 2008), which are known to promote long-term learning gains. Domain-specific educational agents assist with learning in subjects such as science, languages, or professional development for specific domains.

However, beyond general risks such as safety, hallucinations, and bias, the responses of the current state-of-the-art models are often not grade-appropriate (Srivatsa et al., 2025), may diverge from the learning path, conflate user roles, or enter conversational loops (amplified in multi-agent settings) (Li et al., 2023; Chu et al., 2025c; Tran et al., 2025). In summary, this research direction holds huge promise, but key limitations must be ad-

dressed when deploying these systems in sensitive domains like education.

Educational Human-LLM Collaborative Systems: Human-LLM collaborative systems leverage the complementary strengths of humans and LLMs to improve performance in tasks such as data annotation, problem-solving, and decision-making across domains like education and healthcare (Yang et al., 2024; Fragiadakis et al., 2024). In education, LLM-powered systems have been deployed to support both single-turn interactions (e.g., answering questions, explaining steps) (Gao et al., 2024; Hashir et al., 2024) and multi-turn interactions (e.g., Tutor-Copilot (Wang et al., 2024e), GPTeach (Markel et al., 2023)). These systems can, among other things, deepen learner understanding and assist novice tutors in improving their teaching skills and qualities. They are not free from challenges, though. Such systems often lack interpretability, making it hard to trust AI outputs (Yang et al., 2024). They may prioritize correctness over pedagogical goals like conceptual understanding and learner support (Macina et al., 2023c). LLMs also struggle with ambiguity, personalization, and maintaining context in extended interactions, and they rarely offer adaptive feedback tailored to learners’ evolving needs or emotions (Maurya et al., 2025). Addressing these challenges is essential for building reliable and effective educational human-LLM collaborative systems.

Educational Value Alignment: Alignment with human preferences is a key driver to the success of state-of-the-art LLMs (Ouyang et al., 2022; Yao et al., 2023). This ranges from the development of general values-based LLMs (Guo et al., 2025; Team et al., 2024) to models tailored to specific age groups or domains (Nayeem and Rafiei, 2024; Chen et al., 2023). These advancements have also significantly influenced the educational domain, leading to the development of education-specific LLMs such as LearnLM (Team et al., 2024; Jurénka et al., 2024) and pedagogical tutors (Dinucujianu et al., 2025). These LLMs are grounded in pedagogical values (Team et al., 2024; Maurya et al., 2025) and draw on decades of research in the learning sciences to generate pedagogically rich datasets, which are subsequently used for instruction tuning and fine-tuning. These specialized models have proven effective across a wide range of educational applications.

However, an open research question remains –

“What should we align with?” (Yao et al., 2023), which directly affects LLM performance. Specifically, in the case of tutor LLMs, there is currently no consensus among researchers regarding the key pedagogical principles and associated teacher moves that lead to effective learning (Team et al., 2024). Future research should explore the core educational values that need to be integrated to enable the development of more effective educational models.

6 Ongoing Challenges

So far, we have seen how LLMs enabled existing NLP research on educational applications, and also paved way for new use cases. With the growth in their usage in real-world educational scenarios and the potential for personalized education, a discussion about the challenges involved becomes inevitable. In this section, we discuss some of the technical as well as broader application-related challenges in this area. While this discussion about the challenges is non-exhaustive, it broadly highlights some of the general task-agnostic issues related to the use of LLMs and NLP in education.

6.1 Datasets

A lot of NLP research on educational applications relies on the existence of labeled datasets. For most of the tasks, such datasets are created by repurposing existing online resources (e.g., using Wikipedia and Simple Wikipedia, or websites such as Newsela for automatic text simplification), and this is not an exception compared to other areas of research in NLP. Carefully crafted datasets that are specifically developed for a particular task (e.g., grammatical error detection) are not rare, but hard to develop on a large scale. Datasets that consider target user input (e.g., those that contain learner feedback or outcome information) are even rarer. Adding multilingual support to the mix makes dataset development across educational NLP tasks still more challenging.

Although LLMs could offer better zero-shot, off-the-shelf performance for many tasks and languages today, and synthetic data generation with LLMs can address the data scarcity across languages to some extent, we would still need concerted efforts to build high quality educational datasets to develop and evaluate LLMs that can provide educational support across languages. Some recent research also reports poorer performance

of LLMs across four education-related tasks beyond English and recommends verifying the LLM performance in the target language before deployment (Gupta et al., 2025). Imperial et al. (2025)’s recent effort to consolidate multilingual language proficiency assessment datasets under one unified format and license is a welcome step in this direction.

6.2 Pedagogical Evaluation

Across different NLP tasks involving the use of LLMs in the education domain, evaluation challenges have been widely discussed, along with a comparison between automated and human evaluation (Horbach et al., 2020; Vásquez-Rodríguez et al., 2021; Agrawal and Carpuat, 2024; Kobayashi et al., 2024). While most of the discussions around evaluation focused on the task-specific aspects, for technologies such as tutoring systems, a multi-dimensional view of evaluation is necessary.

Traditional evaluation of teacher effectiveness has relied on artifacts, portfolios, self-reports, and student feedback (Goe et al., 2008). More recently, text generation metrics are being explored to assess ITS or AI tutor responses. However, while effective for measuring coherence and fluency, these domain-agnostic metrics often miss deeper pedagogical aspects, depend on gold references, and can be gamed by generic responses (Tack and Piech, 2022). Efforts to capture pedagogical effectiveness more directly have included human evaluations and tailored frameworks defining specific strategies, but these face challenges such as subjectivity, lack of standardization, and limited scope (Jurenka et al., 2024). Some potential directions for evaluating tutor responses were recently proposed by Maurya et al. (2025), where a unified taxonomy was introduced to measure the quality and appropriateness of these responses. However, these rely on human evaluation, which is non-scalable and is typically conducted at the utterance level rather than the conversation level. Naeem et al. (2026a) introduces an automated evaluation model to mitigate reliance on human annotation, leveraging a lightweight multi-task LoRA-based fine-tuning approach on a 2B-parameter model, however, it only captures a limited subset of evaluation dimensions. So far, using LLMs as proxy judges shows promise but still falls short in reliably evaluating complex pedagogical traits (Gu et al., 2024).

Overall, NLP research is understandably model-focused, and that impacts the way we evaluate. But,

user-focused evaluations are also emerging. For example, some recent research points to the mismatch between the user needs and model availability in the context of graded content generation (Asthana et al., 2024; Kim et al., 2024b). Considering the increasing widespread real-world usage of LLMs in educational applications, NLP research could consider a user-first rather than a dataset- and model-first approaches in developing standardized evaluation methodologies for educational applications in future.

6.3 Ethical Issues

We found the discussion around the ethics of using LLMs in education emerging only recently in the NLP community (e.g., Härmäläinen, 2024), yet there has been some thought in this direction in the broader education technology and assessment community. For example, Yan et al. (2024) discuss the ethical implications of the increasing use of LLMs in education considering a range of use cases, and identify transparency for educational stakeholders (teachers, students, parents), privacy, support across languages, and fairness across population groups as the main ethical concerns surrounding the use of LLMs in education, calling for better reporting standards from empirical research that uses LLMs to develop new solutions. This issue of reporting standards is perhaps of most direct relevance to the NLP community.

From an assessment perspective, some recent work discussed the implications of the usage of LLMs in education for academic integrity (de Winter, 2024; Leppänen et al., 2025) and fairness (Yamashita, 2025), and the language assessment community calls for a collaboration between model developers, test creators and subject matter experts, psychometricians and the AI research community to develop education-specific standards for using AI in assessment to ensure reliability and fairness (Bolender et al., 2023; Voss et al., 2023; Xi, 2023).

Hallucination is a well-known concern with LLMs, and educational use cases are not immune to that. Some recent research on text simplification (Hewett et al., 2024; Zečević et al., 2024) pointed to how the tendency to hallucinate increases as the task gets more complex such as generating in a specific domain or in a new language, for example.

7 Conclusions

Our study on how LLMs are integrated into existing research on the NLP-driven educational applications shows that LLMs lead to several interesting new developments which hold a lot of promise for the future in terms of both effective performance as well as inclusive development of applications addressing different languages and population groups. However, there are also several ongoing challenges related to available data, evaluation, and ethical concerns. As suggested by others, we envision an increase in inter-disciplinary collaboration between NLP researchers, domain experts, and educators leading to the development of better assistive and assessment technologies to support students and teachers in the future. We hope that this paper would serve as a good starting point for NLP researchers, providing them with the necessary background on the state of the art in the educational applications of NLP using LLMs and on what lies ahead.

Limitations

We perceive two primary limitations: (a) Since our goal in this paper is to provide an overview of what lies ahead, we did not do an exhaustive survey of the current state of the art, focusing largely on post-LLM research and pointing to relevant surveys for pre-LLM approaches; and (b) We have also primarily restricted ourselves to NLP publication venues, citing research from other related disciplines to a much smaller extent. Our observations and conclusions drawn in this paper should be considered along with these limitations. However, we provide an extensive, although by no means an exhaustive, list of additional readings grouped by the four dimensions – writing, speaking, reading, and tutoring – in the Appendix Tables 1–4, for those interested in exploring these topics further.

Ethical Considerations

The study does not involve the use of any datasets with ethical concerns or training of AI models with potential ethical issues. Hence, we do not anticipate any significant risks associated with this work.

Acknowledgements

Stefano Banno's research is supported by Cambridge University Press & Assessment, a department of The Chancellor, Masters, and Scholars

of the University of Cambridge. Kaushal Kumar Maurya and Ekaterina Kochmar are grateful to the Google Academic Research Award (GARA) 2024 for supporting this research.

References

- Sweta Agrawal and Marine Carpuat. 2024. [Do Text Simplification Systems Preserve Meaning? A Human Evaluation via Reading Comprehension](#). *Transactions of the Association for Computational Linguistics*, 12:432–448.
- Said Al Faraby, Adiwijaya Adiwijaya, and Ade Romadhony. 2024a. Review on neural question generation for education purposes. *International Journal of Artificial Intelligence in Education*, 34(3):1008–1045.
- Said Al Faraby, Ade Romadhony, and 1 others. 2024b. Analysis of LLMs for educational question classification and generation. *Computers and Education: Artificial Intelligence*, 7:100298.
- Bashar Alhafni and Nizar Habash. 2025. [Enhancing text editing for grammatical error correction: Arabic as a case study](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17892–17914, Vienna, Austria. Association for Computational Linguistics.
- Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. [The SAMER Arabic text simplification corpus](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.
- Bashar Alhafni, Go Inoue, Christian Khairallah, and Nizar Habash. 2023. [Advancements in Arabic grammatical error detection and correction: An empirical investigation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6430–6448, Singapore. Association for Computational Linguistics.
- 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).
- Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. [Learning how to simplify from explicit labeling of complex-simplified text pairs](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. [The \(Un\)Suitability of Automatic Evaluation Metrics for Text Simplification](#). *Computational Linguistics*, 47(4):861–889.
- Fernando Alva-Manchego, Regina Stodden, Joseph Marvin Imperial, Abdullah Barayan, Kai North, and Harish Tayyar Madabushi. 2025. [Findings of the TSAR 2025 shared task on readability-controlled text simplification](#). In *Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025)*, pages 116–130, Suzhou, China. Association for Computational Linguistics.
- John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An integrated theory of the mind. *Psychological review*, 111(4):1036.
- John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.
- Vahid Aryadoust, Azrifah Zakaria, and Yichen Jia. 2024. Investigating the affordances of OpenAI’s large language model in developing listening assessments. *Computers and Education: Artificial Intelligence*, 6:100204.
- Sumit Asthana, Hannah Rashkin, Elizabeth Clark, Fantine Huot, and Mirella Lapata. 2024. [Evaluating LLMs for targeted concept simplification for domain-specific texts](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6208–6226, Miami, Florida, USA. Association for Computational Linguistics.
- John Atkinson and Diego Palma. 2025. An LLM-based hybrid approach for enhanced automated essay scoring. *Scientific Reports*, 15(1):14551.
- 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.
- Stefano Banno, Hari Krishna Vydana, Kate Knill, and Mark Gales. 2024. [Can GPT-4 do L2 analytic assessment?](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 149–164, Mexico City, Mexico. Association for Computational Linguistics.
- Stefano Bannò, Kate Knill, and Mark Gales. 2025a. [Exploiting the English Vocabulary Profile for L2 word-level vocabulary assessment with LLMs](#). *Preprint*, arXiv:2506.02758.

- Stefano Bannò, Katherine M Knill, Marco Matassoni, Vyas Raina, and Mark Gales. 2023a. [Assessment of L2 Oral Proficiency Using Self-Supervised Speech Representation Learning](#). In *9th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 126–130.
- Stefano Bannò, Rao Ma, Mengjie Qian, Kate M. Knill, and Mark J. F. Gales. 2024a. [Towards End-to-End Spoken Grammatical Error Correction](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10791–10795.
- Stefano Bannò, Rao Ma, Mengjie Qian, Siyuan Tang, Kate Knill, and Mark Gales. 2025b. [Natural Language-based Assessment of L2 Oral Proficiency using LLMs](#). Preprint, arXiv:2507.10200.
- Stefano Bannò, Michela Rais, and Marco Matassoni. 2023b. [Grammatical Error Correction for L2 Speech Using Publicly Available Data](#). In *9th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 136–140.
- Stefano Bannò, Hari Krishna Vydana, Kate Knill, and Mark Gales. 2024b. [Can GPT-4 do L2 analytic assessment?](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 149–164, Mexico City, Mexico. Association for Computational Linguistics.
- Abdullah Barayan, Jose Camacho-Collados, and Fernando Alva-Manchego. 2025. [Analysing zero-shot readability-controlled sentence simplification](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6762–6781, Abu Dhabi, UAE. Association for Computational Linguistics.
- May Bashendy, Walid Massoud, Sohaila Eltanbouly, Salam Albatarni, Marwan Sayed, Abrar Abir, Houda Bouamor, and Tamer Elsayed. 2026. [LAILA: A large trait-based dataset for Arabic automated essay scoring](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3074–3091, Rabat, Morocco. Association for Computational Linguistics.
- Claudia Baur, Andrew Caines, Cathy Chua, Johanna Gerlach, Mengjie Qian, Helmer Rayner, and Xizi Wei. 2019. [Overview of the 2019 Spoken CALL Shared Task](#). In *8th ISCA Workshop on Speech and Language Technology in Education (SLaTE 2019)*. ISCA.
- Claudia Baur, Andrew Caines, Cathy Chua, Johanna Gerlach, Mengjie Qian, Manny Rayner, Martin Russell, Helmer Strik, and Xizi Wei. 2018. [Overview of the 2018 spoken CALL shared task](#). In *Proceedings of Interspeech 2018*, pages 2354–2358.
- Claudia Baur, Cathy Chua, Johanna Gerlach, Emmanuel Rayner, Martin Russel, Helmer Strik, and Xizi Wei. 2017. [Overview of the 2017 spoken CALL shared task](#). In *Proceedings of the 7th Workshop on Speech and Language Technology in Education (SLaTE 2017)*.
- Beata Beigman Klebanov and Nitin Madnani. 2020. [Automated evaluation of writing – 50 years and counting](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7796–7810, Online. Association for Computational Linguistics.
- Beata Beigman Klebanov and Nitin Madnani. 2022. *Automated essay scoring*. Springer Nature.
- Riadh Belkebir and Nizar Habash. 2021. [Automatic error type annotation for Arabic](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606, Online. Association for Computational Linguistics.
- Jared Bernstein, Michael Cohen, Hy Murveit, Dmitry Rtischev, and Mitchel Weintraub. 1990. [Automatic evaluation and training in English pronunciation](#). In *First International Conference on Spoken Language Processing*.
- Ummugul Bezirhan and Matthias von Davier. 2023. Automated reading passage generation with OpenAI’s large language model. *Computers and Education: Artificial Intelligence*, 5:100161.
- 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.
- Benjamin S Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16.
- Brad Bolender, Charles Foster, and Sara Vispoel. 2023. The criticality of implementing principled design when using AI technologies in test development. *Language Assessment Quarterly*, 20(4-5):512–519.
- Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. [The MERLIN corpus: Learner language and the CEFR](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Christopher G Brinton, Ruediger Rill, Sangtae Ha, Mung Chiang, Robert Smith, and William Ju. 2014. Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Transactions on Learning Technologies*, 8(1):136–148.
- 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*, pages 643–701.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International journal of artificial intelligence in education*, 25(1):60–117.
- 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.
- Jill Burstein. 2009. Opportunities for natural language processing research in education. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 6–27. Springer.
- Heather Buzick, Maria Elena Oliveri, Yigal Attali, and Michael Flor. 2016. Comparing human and automated essay scoring for prospective graduate students with learning disabilities and/or ADHD. *Applied Measurement in Education*, 29(3):161–172.
- Yida Cai, Kun Liang, Sanwoo Lee, Qinghan Wang, and Yunfang Wu. 2025. [Rank-Then-Score: Enhancing Large Language Models for Automated Essay Scoring](#). *Preprint*, arXiv:2504.05736.
- Andrew Caines, Luca Benedetto, Shiva Taslimipoor, Christopher Davis, Yuan Gao, Øistein E Andersen, Zheng Yuan, Mark Elliott, Russell Moore, Christopher Bryant, and 1 others. 2023. On the Application of Large Language Models for Language Teaching and Assessment Technology. In *LLM@ AIED*.
- Andrew Caines, Christian Bentz, Calbert Graham, Tim Polzehl, and Paula Buttery. 2016. [Crowdsourcing a multi-lingual speech corpus: Recording, transcription and annotation of the CrowdIS corpora](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 2145–2152, Portorož, Slovenia. European Language Resources Association (ELRA).
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. [The teacher-student chat-room corpus](#). In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20, Gothenburg, Sweden. LiU Electronic Press.
- Rémi Cardon, Adrien Bibal, Rodrigo Wilkens, David Alfter, Magali Norré, Adeline Müller, Watrin Patrick, and Thomas François. 2022. [Linguistic corpus annotation for automatic text simplification evaluation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1842–1866, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. [Motivations and methods for text simplification](#). In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.
- Naina Chaturvedi. 2024. [LLMs and NLP for generalized learning in AI-enhanced educational videos and powering curated videos with generative intelligence](#). In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pages 148–154, Miami, FL, USA. Association for Computational Linguistics.
- Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018a. LearningQ: a large-scale dataset for educational question generation. In *Proceedings of the international AAAI conference on web and social media*, volume 12.
- Hongbo Chen and Ben He. 2013. [Automated essay scoring by maximizing human-machine agreement](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, Seattle, Washington, USA. Association for Computational Linguistics.
- Lei Chen, Jidong Tao, Shabnam Ghaffarzadegan, and Yao Qian. 2018b. [End-to-end neural network based automated speech scoring](#). In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, pages 6234–6238.
- Yuyan Chen, Chenwei Wu, Songzhou Yan, Panjun Liu, and Yanghua Xiao. 2024. [Dr.Academy: A benchmark for evaluating questioning capability in education for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3138–3167, Bangkok, Thailand. 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, and 1 others. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

- Alison Chi, Li-Kuang Chen, Yi-Chen Chang, Shu-Hui Lee, and Jason S. Chang. 2023. [Learning to Paraphrase Sentences to Different Complexity Levels](#). *Transactions of the Association for Computational Linguistics*, 11:1332–1354.
- 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, Prague, Czech Republic. Association for Computational Linguistics.
- Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. [Classifying syntactic errors in learner language](#). In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107, Online. Association for Computational Linguistics.
- Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. 2024. [AutoTutor meets Large Language Models: A Language Model Tutor with Rich Pedagogy and Guardrails](#). *Preprint*, arXiv:2402.09216.
- Bryan R Christ, Jonathan Kropko, and Thomas Hartvigsen. 2024. [MATHWELL: Generating educational math word problems using teacher annotations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11914–11938, Miami, Florida, USA. Association for Computational Linguistics.
- SeongYeub Chu, Jong Woo Kim, Bryan Wong, and Mun Yong Yi. 2025a. [Rationale behind essay scores: Enhancing S-LLM’s multi-trait essay scoring with rationale generated by LLMs](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5796–5814, Albuquerque, New Mexico. Association for Computational Linguistics.
- Yun-Hsin Chu, Shuai Zhu, Shou-Yi Hung, Bo-Ting Lin, En-Shiun Annie Lee, and Richard Tzong-Han Tsai. 2025b. [ATAIGI: An AI-Powered Multimodal Learning App Leveraging Generative Models for Low-Resource Taiwanese Hokkien](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 11–19.
- Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. [Qwen2-audio technical report](#). *arXiv preprint arXiv:2407.10759*.
- Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, and 1 others. 2025c. [LLM agents for education: Advances and applications](#). *arXiv preprint arXiv:2503.11733*.
- Zhendong Chu, Jian Xie, Shen Wang, Zichao Wang, and Qingsong Wen. 2025d. [UniEDU: Toward unified and efficient large multimodal models for educational tasks](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1007–1016, Suzhou (China). Association for Computational Linguistics.
- William J. Clancey. 1987. [Knowledge-Based Tutoring: The GUIDON Program](#).
- Kevyn Collins-Thompson. 2014. [Computational assessment of text readability: A survey of current and future research](#). *ITL - International Journal of Applied Linguistics*, 165(2):97–135.
- Kevyn Collins-Thompson and James P. Callan. 2004. [A language modeling approach to predicting reading difficulty](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 193–200, Boston, Massachusetts, USA. Association for Computational Linguistics.
- William Coster and David Kauchak. 2011. [Simple English Wikipedia: A new text simplification task](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.
- Sylvain Coulange, Marie-Hélène Fries, Monica Masperi, and Solange Rossato. 2024. [A corpus of spontaneous L2 English speech for real-situation speaking assessment](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 293–297, Torino, Italia. ELRA and ICCL.
- Steven Coyne, Diana Galvan-Sosa, Ryan Spring, Camélia Guerraoui, Michael Zock, Keisuke Sakaguchi, and Kentaro Inui. 2025. [Annotating errors in english learners’ written language production: Advancing automated written feedback systems](#). In *Artificial Intelligence in Education: 26th International Conference, AIED 2025, Palermo, Italy, July 22–26, 2025, Proceedings, Part IV*, page 292–306, Berlin, Heidelberg. Springer-Verlag.
- 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](#). *Preprint*, arXiv:2303.14342.
- 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.
- Scott Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Youngmeen Kim, Wesley Morris, Meg Benner, Aigner Picou, and Ulrich Boser. 2023. [The English Language Learning Insight, Proficiency and Skills Evaluation \(ELLIPSE\) corpus](#). *International Journal of Learner Corpus Research*, 9(2):248–269.

- Catia Cucchiarini, Helmer Strik, and Lou Boves. 1997. [Automatic evaluation of Dutch pronunciation by using speech recognition technology](#). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 622–629.
- Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. [Stepwise verification and remediation of student reasoning errors with large language model tutors](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. [Building a large annotated corpus of learner English: The NUS corpus of learner English](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- 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, Montréal, Canada. Association for Computational Linguistics.
- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipour, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.
- 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*, pages 328–334, New Orleans, Louisiana. Association for Computational Linguistics.
- Joost CF de Winter. 2024. Can ChatGPT pass high school exams on English language comprehension? *International Journal of Artificial Intelligence in Education*, 34(3):915–930.
- Juan Dempere, Kennedy Modugu, Allam Hesham, and Lakshmana Kumar Ramasamy. 2023. The impact of ChatGPT on higher education. In *Frontiers in Education*, volume 8, page 1206936. Frontiers Media SA.
- 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, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. [Measuring conversational uptake: A case study on student-teacher interactions](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1638–1653, Online. Association for Computational Linguistics.
- Paul Denny, Sumit Gulwani, Neil T. Heffernan, Tanja Käser, Steven Moore, Anna N. Rafferty, and Adish Singla. 2024. [Generative AI for Education \(GAIED\): Advances, Opportunities, and Challenges](#). *Preprint*, arXiv:2402.01580.
- Aniket Deroy, Subhankar Maity, and Sudeshna Sarkar. 2025. [MIRROR: A Novel Approach for the Automated Evaluation of Open-Ended Question Generation](#). In *Proceedings of Large Foundation Models for Educational Assessment*, volume 264 of *Proceedings of Machine Learning Research*, pages 3–32. PMLR.
- Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. [Linguistic features for readability assessment](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17, Seattle, WA, USA → Online. Association for Computational Linguistics.
- David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. From Problem-Solving to Teaching Problem-Solving: Aligning LLMs with Pedagogy using Reinforcement Learning. *arXiv preprint arXiv:2505.15607*.
- Sidney D’Mello, Andrew Olney, Claire Williams, and Patrick Hays. 2012. [Gaze tutor: A gaze-reactive intelligent tutoring system](#). *International Journal of Human-Computer Studies*, 70(5):377–398.
- Myroslava O. Dzikovska, Johanna D. Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B. Callaway. 2010. [Beetle II: A system for tutoring and computational linguistics experimentation](#). In *Proceedings of the ACL 2010 System Demonstrations*, pages 13–18, Uppsala, Sweden. Association for Computational Linguistics.
- Yo Ehara. 2018. Building an english vocabulary knowledge dataset of japanese english-as-a-second-language learners using crowdsourcing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2012. Mining words in the minds of second language learners: Learner-specific word difficulty. In *Proceedings of COLING 2012*, pages 799–814.

- Noemie Elhadad and Komal Sutaria. 2007. [Mining a lexicon of technical terms and lay equivalents](#). In *Biological, translational, and clinical language processing*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics.
- Sabina Elkins, Ekaterina Kochmar, Jackie CK Cheung, and Iulian Serban. 2024. How Teachers Can Use Large Language Models and Bloom’s Taxonomy to Create Educational Quizzes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23084–23091.
- Sabina Elkins, Ekaterina Kochmar, Iulian Serban, and Jackie CK Cheung. 2023. How useful are educational questions generated by large language models? In *International Conference on Artificial Intelligence in Education*, pages 536–542. Springer.
- Björn Engelmann, Christin Katharina Kreutz, Fabian Haak, and Philipp Schaer. 2024. [ARTS: Assessing readability & text simplicity](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14925–14942, Miami, Florida, USA. Association for Computational Linguistics.
- Sean B Eom, H Joseph Wen, and Nicholas Ashill. 2006. The determinants of students’ perceived learning outcomes and satisfaction in university online education: An empirical investigation. *Decision Sciences Journal of Innovative Education*, 4(2):215–235.
- Keelan Evanini, Matthew Mulholland, Rutuja Ubale, Yao Qian, Robert A Pugh, Vikram Ramanarayanan, and Aoife Cahill. 2018. [Improvements to an Automated Content Scoring System for Spoken CALL Responses: the ETS Submission to the Second Spoken CALL Shared Task](#). In *Proc. Interspeech 2018*, pages 2379–2383.
- Marc Evers and Anton Nijholt. 2000. Jacob - An Animated Instruction Agent in Virtual Reality. In *Advances in Multimodal Interfaces — ICMI 2000*, pages 526–533, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Dengzhao Fang, Jipeng Qiang, Xiaoye Ouyang, Yi Zhu, Yunhao Yuan, and Yun Li. 2025. [Collaborative document simplification using multi-agent systems](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 897–912, Abu Dhabi, UAE. Association for Computational Linguistics.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jipeng Hu, Lidia S. Chao, and Yue Zhang. 2023. [Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation](#). *Preprint*, arXiv:2304.01746.
- Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. [Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339, Torino, Italia. ELRA and ICCL.
- Menna Fateen and Tsunenori Mine. 2024. Developing a tutoring dialog dataset to optimize LLMs for educational use. *arXiv preprint arXiv:2410.19231*.
- 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*, pages 578–587, Denver, Colorado. Association for Computational Linguistics.
- Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. [Grammatical error correction using hybrid systems and type filtering](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.
- Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. [A comparison of features for automatic readability assessment](#). In *Coling 2010: Posters*, pages 276–284, Beijing, China. Coling 2010 Organizing Committee.
- Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. 2025. [Automatic Short Answer Grading in the LLM Era: Does GPT-4 with Prompt Engineering beat Traditional Models?](#) In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK ’25*, page 93–103, New York, NY, USA. Association for Computing Machinery.
- Michael Flor. 2025. *Automatic Question Generation*. Springer, Switzerland.
- George Fragiadakis, Christos Diou, George Kousiouris, and Mara Nikolaidou. 2024. Evaluating human-AI collaboration: A review and methodological framework. *arXiv preprint arXiv:2407.19098*.
- Horacio Franco, Victor Abrash, Kristin Precoda, Harry Bratt, Ramana Rao, John Butzberger, Romain Rossier, and Federico Cesari. 2000. [The SRI EduS-peak™ system: Recognition and pronunciation scoring for language learning](#). In *Proceedings of InSTILL*, pages 123–128.
- Reva Freedman. 2000. [Plan-based dialogue management in a physics tutor](#). In *Sixth Applied Natural Language Processing Conference*, pages 52–59, Seattle, Washington, USA. Association for Computational Linguistics.
- Kaiqi Fu, Linkai Peng, Nan Yang, and Shuran Zhou. 2024. [Pronunciation Assessment with Multi-modal Large Language Models](#). *Preprint*, arXiv:2407.09209.

- Kevin Fuchs. 2023. Exploring the opportunities and challenges of NLP models in higher education: is Chat GPT a blessing or a curse? In *Frontiers in education*, volume 8, page 1166682. Frontiers Media SA.
- Peizhong Gao, Ao Xie, Shaoguang Mao, Wenshan Wu, Yan Xia, Haipeng Mi, and Furu Wei. 2024. Meta reasoning for large language models. *arXiv preprint arXiv:2406.11698*.
- Yingqiang Gao, Kaede Johnson, David Froehlich, Luisa Carrer, and Sarah Ebling. 2025. [Evaluating the Effectiveness of Direct Preference Optimization for Personalizing German Automatic Text Simplifications for Persons with Intellectual Disabilities](#). *Preprint*, arXiv:2507.01479.
- 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.
- Aivars Glaznieks, Jennifer-Carmen Frey, Andrea Abel, Lionel Nicolas, and Chiara Vettori. 2023. The Kolipsi corpus Family: Resources for learner corpus research in Italian and German. *IJCoL. Italian Journal of Computational Linguistics*, 9(9-2).
- Laura Goe, Courtney Bell, and Olivia Little. 2008. Approaches to Evaluating Teacher Effectiveness: A Research Synthesis. *National Comprehensive Center for Teacher Quality*.
- 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.
- 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.
- Guher Gorgun and Okan Bulut. 2024. [Current Evaluation Methods are a Bottleneck in Automatic Question Generation](#). In *Proceedings of the 2024 AAAI Conference on Artificial Intelligence*, volume 257 of *Proceedings of Machine Learning Research*, pages 3–8. PMLR.
- Arthur C. Graesser, G Tanner Jackson, Hyun-Jeong Joyce Kim, and Andrew Olney. 2006. AutoTutor 3-D Simulations: Analyzing Users’ Actions and Learning Trends. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.
- Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. 2004. [AutoTutor: A tutor with dialogue in natural language](#). *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192.
- Arthur C. Graesser, Kurt VanLehn, Carolyn P Rosé, Pamela W Jordan, and Derek Harter. 2001. Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39–39.
- Arthur C. Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Roger Kreuz, Tutoring Research Group, and 1 others. 1999. AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, 1(1):35–51.
- Sylviane Granger, Estelle Dagneaux, Fanny Meunier, Magali Paquot, and 1 others. 1993. The international corpus of learner English. *English language corpora: Design, analysis and exploitation*, pages 57–71.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594*.
- Theo Guidroz, Diego Ardila, Jimmy Li, Adam Mansour, Paul Jhun, Nina Gonzalez, Xiang Ji, Mike Sanchez, Sujay Kakarmath, Mathias MJ Bellaiche, and 1 others. 2025. LLM-based Text Simplification and its Effect on User Comprehension and Cognitive Load. *arXiv preprint arXiv:2505.01980*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. [Dynamic multi-level multi-task learning for sentence simplification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Vansh Gupta, Sankalan Pal Chowdhury, Vilém Zouhar, Donya Rooein, and Mrinmaya Sachan. 2025. Multilingual Performance Biases of Large Language Models in Education. *arXiv preprint arXiv:2504.17720*.
- Nizar Habash and David Palfreyman. 2022. [ZAEBUC: An annotated Arabic-English bilingual writer corpus](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 79–88, Marseille, France. European Language Resources Association.
- Mika Härmäläinen. 2024. [Legal and ethical considerations that hinder the use of LLMs in a Finnish institution of higher education](#). In *Proceedings of the Workshop on Legal and Ethical Issues in Human Language Technologies @ LREC-COLING 2024*, pages 24–27, Torino, Italia. ELRA and ICCL.

- 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.
- Michael Hardy and Yunsung Kim. 2026. Knowledge without Wisdom: Measuring Misalignment between LLMs and Intended Impact. *arXiv preprint arXiv:2603.00883*.
- Mohammad Hashir, Avinash Kumar Sharma, Shah Faisal, and Gourav Rawat. 2024. Automatic Feedback Generation In NLP: A Systematic Review. In *2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N)*, pages 1492–1497. IEEE.
- Akio Hayakawa, Tomoyuki Kajiwar, Hiroki Ouchi, and Taro Watanabe. 2022. [JADES: New text simplification dataset in Japanese targeted at non-native speakers](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 179–187, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.
- Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. [Arabic word-level readability visualization for assisted text simplification](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.
- Neil T Heffernan, Kenneth R Koedinger, and Leena Razzaq. 2008. Expanding the model-tracing architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education*, 18(2):153–178.
- Freya Hewett, Hadi Asghari, and Manfred Stede. 2024. [Elaborative simplification for German-language texts](#). In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 29–39, Kyoto, Japan. Association for Computational Linguistics.
- Kevin Hirschi, Okim Kang, Mu Yang, John H. L. Hansen, and Kyle Beloin. 2025. [Artificial intelligence-generated feedback for second language intelligibility: An exploratory intervention study on effects and perceptions](#). *Language Learning*, 75(S1):204–241.
- Kate S Hone and Ghada R El Said. 2016. Exploring the factors affecting MOOC retention: A survey study. *Computers & Education*, 98:157–168.
- 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, Marseille, France. European Language Resources Association.
- Andrea Horbach, Joey Pehlke, Ronja Laarmann-Quante, and Yuning Ding. 2024. Crosslingual content scoring in five languages using machine-translation and multilingual transformer models. *International Journal of Artificial Intelligence in Education*, 34(4):1294–1320.
- Andrea Horbach, Dirk Scholten-Akoun, Yuning Ding, and Torsten Zesch. 2017. [Fine-grained essay scoring of a complex writing task for native speakers](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 357–366, Copenhagen, Denmark. Association for Computational Linguistics.
- Yichen Huang and Ekaterina Kochmar. 2024. [REF-eree: A REFERENCE-FREE model-based metric for text simplification](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13740–13753, Torino, Italia. ELRA and ICCL.
- Seonjeong Hwang, Yunsu Kim, and Gary Lee. 2024. [Cross-lingual transfer for automatic question generation by learning interrogative structures in target languages](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3194–3208, Miami, Florida, USA. Association for Computational Linguistics.
- William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. [Aligning sentences from standard Wikipedia to Simple Wikipedia](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217, Denver, Colorado. Association for Computational Linguistics.
- Joseph Marvin Imperial, Abdullah Barayan, Regina Stodden, Rodrigo Wilkens, Ricardo Munoz Sanchez, Lingyun Gao, Melissa Torgbi, Dawn Knight, Gail Forey, Reka R. Jablonkai, Ekaterina Kochmar, Robert Reynolds, Eugenio Ribeiro, Horacio Saggion, Elena Volodina, Sowmya Vajjala, Thomas Francois, Fernando Alva-Manchego, and Harish Tayyar Madabushi. 2025. [UniversalCEFR: Enabling Open Multilingual Research on Language Proficiency Assessment](#). *Preprint*, arXiv:2506.01419.
- Joseph Marvin Imperial, Gail Forey, and Harish Tayyar Madabushi. 2024. [Standardize: Aligning language models with expert-defined standards for content generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1573–1594, Miami, Florida, USA. Association for Computational Linguistics.
- Shin’ichiro Ishikawa. 2013. The ICNALE and sophisticated contrastive interlanguage analysis of Asian learners of English. *Learner corpus studies in Asia and the world*, 1:91–118.

- Shin'ichiro Ishikawa. 2019. The ICNALE Spoken Dialogue: A New Dataset for the Study of Asian Learners' Performance in L2 English Interviews. *English teaching*, 74(4):153–177.
- Shin'ichiro Ishikawa. 2014. Design of the ICNALE-Spoken: A new database for multi-modal contrastive interlanguage analysis. *Learner corpus studies in Asia and the world*, 2:63–76.
- Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. [Automatic error detection in the Japanese learners' English spoken data](#). In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 145–148, Sapporo, Japan. Association for Computational Linguistics.
- Emi Izumi, Kiyotaka Utimoto, and Hitoshi Isahara. 2004. The NICT JLE Corpus: Exploiting the language learners' speech database for research and education. *Special Issues of International Journal of the Computer*, 1:31–48.
- Chao Jiang and Wei Xu. 2024. [MedReadMe: A systematic study for fine-grained sentence readability in medical domain](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17293–17319, Miami, Florida, USA. Association for Computational Linguistics.
- Meiqing Jin, Liam Dugan, and Chris Callison-Burch. 2025. [Controlling Difficulty of Generated Text for AI-Assisted Language Learning](#). *Preprint*, arXiv:2506.04072.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. [The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33, Baltimore, Maryland. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. [Phrase-based machine translation is state-of-the-art for automatic grammatical error correction](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Irina Jurenka, Markus Kunesch, Kevin R. McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, Ankit Anand, Miruna Pîslar, Stephanie Chan, Lisa Wang, Jennifer She, Parsa Mahmoudieh, Aliya Rysbek, Wei-Jen Ko, Andrea Huber, and 55 others. 2024. [Towards Responsible Development of Generative AI for Education: An Evaluation-Driven Approach](#). *Preprint*, arXiv:2407.12687.
- Tomoyuki Kajiwaru and Mamoru Komachi. 2016. [Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158, Osaka, Japan. The COLING 2016 Organizing Committee.
- Tomoyuki Kajiwaru, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. [Selecting proper lexical paraphrase for children](#). In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013)*, pages 59–73, Kaohsiung, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. [Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- Masahiro Kaneko and Naoaki Okazaki. 2023. [Reducing sequence length by predicting edit spans with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029, Singapore. Association for Computational Linguistics.
- Masahiro Kaneko and Naoaki Okazaki. 2024. [Controlled generation with prompt insertion for natural language explanations in grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3955–3961, Torino, Italia. ELRA and ICCL.
- Manu Kapur. 2008. Productive Failure. *Cognition and Instruction*, 26(3):379–424.
- Penny Karanasou, Mengjie Qian, Stefano Bannò, Kate Knill, and Mark JF Gales. 2025. Data Augmentation for Spoken Grammatical Error Correction. In *To appear in Proc. 10th Workshop on Speech and Language Technology in Education (SLaTE 2025)*.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, and 1 others. 2023. ChatGPT for good?

- On opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.
- Anisia Katinskaia and Roman Yangarber. 2024. [GPT-3.5 for grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- 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.
- David Kauchak. 2013. [Improving text simplification language modeling using unsimplified text data](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.
- Zixuan Ke and Vincent Ng. 2019. Automated Essay Scoring: A Survey of the State of the Art. In *IJCAI*, volume 19, pages 6300–6308.
- Collins-Thompson Kevyn. 2014. Computational assessment of text readability. *ITL-International Journal of Applied Linguistics*, 165(2):97–135.
- Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023a. [BLESS: Benchmarking Large Language Models on Sentence Simplification](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13291–13309.
- Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023b. [BLESS: Benchmarking large language models on sentence simplification](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13291–13309, Singapore. Association for Computational Linguistics.
- Eesung Kim, Jae-Jin Jeon, Hyeji Seo, and Hoon Kim. 2022. [Automatic Pronunciation Assessment using Self-Supervised Speech Representation Learning](#). In *Proceedings of Interspeech 2022*, pages 1411–1415.
- Haechan Kim, Junho Myung, Seoyoung Kim, Sungpah Lee, Dongyeop Kang, and Juho Kim. 2024a. [LearnerVoice: A Dataset of Non-Native English Learners’ Spontaneous Speech](#). In *Proceedings of Interspeech 2025*, pages 2325–2329.
- Jenia Kim, Stefan Leijnen, and Lisa Beinborn. 2024b. [Considering human interaction and variability in automatic text simplification](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 52–60, Miami, Florida, USA. Association for Computational Linguistics.
- Kate Knill, Diane Nicholls, Mark J F Gales, Mengjie Qian, and Pawel Strojinski. 2024. [Speak & Improve Corpus 2025: an L2 English Speech Corpus for Language Assessment and Feedback](#). *arXiv preprint arXiv:2412.11986*.
- Kate M. Knill, Mark JF Gales, Potsawee P. Manakul, and Andrew P. Caines. 2019. [Automatic Grammatical Error Detection of Non-native Spoken Learner English](#). In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 8127–8131.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Large language models are state-of-the-art evaluator for grammatical error correction](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.
- Ekaterina Kochmar, Øistein Andersen, and Ted Briscoe. 2012. [HOO 2012 error recognition and correction shared task: Cambridge University submission report](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 242–250, Montréal, Canada. Association for Computational Linguistics.
- Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. [Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors](#). *arXiv preprint arXiv:2507.10579*.
- Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. 2025. [Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task](#). *arXiv preprint arXiv:2506.08872*.
- Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. [Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 204–211, Marseille, France. European Language Resources Association.
- Nischal Ashok Kumar and Andrew Lan. 2024. [Improving socratic question generation using data augmentation and preference optimization](#). *arXiv preprint arXiv:2403.00199*.

- 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.
- John P. Lalor, Pedro Rodriguez, Joao Sedoc, and Jose Hernandez-Orallo. 2024. Item response theory for natural language processing. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Malta. Association for Computational Linguistics.
- Thomas K Landauer, D Laham, and PW 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.
- Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405.
- Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. *Automated Grammatical Error Detection for Language Learners*, 2 edition, volume 1 of *Synthesis Lectures on Human Language Technologies*. Springer, Cham.
- Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. [Pushing on text readability assessment: A transformer meets handcrafted linguistic features](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Bruce W. Lee and Jason Lee. 2023. [Prompt-based learning for text readability assessment](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1819–1824, Dubrovnik, Croatia. Association for Computational Linguistics.
- Changyoon Lee, Junho Myung, Jieun Han, Jiho Jin, and Alice Oh. 2023. [Learning from Teaching Assistants to Program with Subgoals: Exploring the Potential for AI Teaching Assistants](#). *Preprint*, arXiv:2309.10419.
- Sungjin Lee, Hyungjong Noh, Kyusong Lee, and Gary Geunbae Lee. 2011. Grammatical error detection for corrective feedback provision in oral conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 797–802.
- Leo Leppänen, Lili Aunimo, Arto Hellas, Jukka K. Nurminen, and Linda Mannila. 2025. [How Large Language Models Are Changing MOOC Essay Answers: A Comparison of Pre- and Post-LLM Responses](#). *Preprint*, arXiv:2504.13038.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative Agents for “Mind” Exploration of Large Language Model Society. *Advances in Neural Information Processing Systems*, 36:51991–52008.
- Shengjie Li and Vincent Ng. 2024a. [Automated essay scoring: A reflection on the state of the art](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.
- Shengjie Li and Vincent Ng. 2024b. Automated essay scoring: Recent successes and future directions. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence, Jeju, Republic of Korea*.
- Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. 2024a. Socraticlm: Exploring socratic personalized teaching with large language models. *Advances in Neural Information Processing Systems*, 37:85693–85721.
- Tianyi Liu, Julia Chatain, Laura Kobel-Keller, Gerd Kortemeyer, Thomas Willwacher, and Mrinmaya Sachan. 2024b. [AI-assisted Automated Short Answer Grading of Handwritten University Level Mathematics Exams](#). *Preprint*, arXiv:2408.11728.
- Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. [LLM comparative assessment: Zero-shot NLG evaluation through pairwise comparisons using large language models](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian’s, Malta. 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.
- S. Magalí López Cortez, Mark Josef Norris, and Steve Duman. 2024. [GMEG-EXP: A dataset of human- and LLM-generated explanations of grammatical and fluency edits](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7785–7800, Torino, Italia. ELRA and ICCL.
- Hao-Chien Lu, Jhen-Ke Lin, Hong-Yun Lin, Chung-Chun Wang, and Berlin Chen. 2025. [Advancing automated speaking assessment leveraging multifaceted relevance and grammar information](#). *Preprint*, arXiv:2506.16285.
- Yiting Lu, Stefano Bannò, and Mark Gales. 2022. [On assessing and developing spoken ‘grammatical error correction’ systems](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 51–60,

- Seattle, Washington. Association for Computational Linguistics.
- Yiting Lu, Mark J. F. Gales, Katherine M. Knill, Potsawee Manakul, and Yu Wang. 2019a. [Disfluency detection for spoken learner English](#). In *Proceedings of the 8th Workshop on Speech and Language Technology for Education (SLaTE)*, pages 74–78.
- Yiting Lu, Mark J. F. Gales, and Yu Wang. 2020. [Spoken Language ‘Grammatical Error Correction’](#). *Proceedings of Interspeech 2020*, pages 3840–3844.
- Yiting Lu, Mark JF Gales, Kate M. Knill, Potsawee Manakul, Linlin Wang, and Yu Wang. 2019b. [Impact of ASR Performance on Spoken Grammatical Error Detection](#). *Proceedings of Interspeech 2019*, pages 1876–1880.
- Haohao Luo, Yang Deng, Ying Shen, See-Kiong Ng, and Tat-Seng Chua. 2024. [Chain-of-exemplar: Enhancing distractor generation for multimodal educational question generation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7978–7993, Bangkok, Thailand. Association for Computational Linguistics.
- Rao Ma, Mengjie Qian, Siyuan Tang, Stefano Bannò, Kate M. Knill, and Mark J. F. Gales. 2025. [Assessment of L2 Oral Proficiency using Speech Large Language Models](#). *Preprint*, arXiv:2505.21148.
- Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023a. [MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.
- Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. [Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems](#). *arXiv preprint arXiv:2305.14536*.
- Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023c. [Opportunities and challenges in neural dialog tutoring](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2357–2372, Dubrovnik, Croatia. 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.
- Mounica Maddela and Wei Xu. 2018. [A word-complexity lexicon and a neural readability ranking model for lexical simplification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760, Brussels, Belgium. Association for Computational Linguistics.
- Subhankar Maity and Aniket Deroy. 2025. [Leveraging Prompt-Tuning for Bengali Grammatical Error Explanation Using Large Language Models](#). *arXiv preprint arXiv:2504.05642*.
- Rizwaan Malik, Dorna Abdi, Rose Wang, and Dorottya Demszky. 2025. [Scaffolding middle school mathematics curricula with large language models](#). *British Journal of Educational Technology*.
- Andrey Malinin, Anton Ragni, Kate Knill, and Mark Gales. 2017. [Incorporating Uncertainty into Deep Learning for Spoken Language Assessment](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 45–50.
- Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. [EdiT5: Semi-autoregressive text editing with t5 warm-start](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2126–2138, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. [FELIX: Flexible text editing through tagging and insertion](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. [Encode, tag, realize: High-precision text editing](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.
- Jeziel Marinho, Rafael Anchiêta, and Raimundo Moura. 2021. [Essay-BR: a Brazilian Corpus of Essays](#). In *Anais do III Dataset Showcase Workshop*, pages 53–64, Porto Alegre, RS, Brasil. SBC.
- Julia M. Markel, Steven G. Opferman, James A. Landay, and Chris Piech. 2023. [GPTeach: Interactive TA Training with GPT-based Students](#). In *Proceedings of the Tenth ACM Conference on Learning @ Scale, L@S ’23*, page 226–236, New York, NY, USA. Association for Computing Machinery.
- Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. [Controllable sentence simplification](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

- Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. [MUSS: Multilingual unsupervised sentence simplification by mining paraphrases](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfah, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025. [The MultiGEC-2025 shared task on multilingual grammatical error correction at NLP4CALL](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 1–33, Tallinn, Estonia. University of Tartu Library.
- 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).
- Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. [Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.
- Simon W McKnight, Arda Civelekoglu, Mark Gales, Stefano Bannò, Adian Liusie, and Katherine M Knill. 2023. [Automatic Assessment of Conversational Speaking Tests](#). In *9th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 99–103.
- Wolfgang Menzel, Eric Atwell, Patrizia Bonaventura, Daniel Herron, Peter Howarth, Rachel Morton, and Clive Souter. 2000. [The ISLE corpus of non-native spoken English](#). In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).
- Stuart Mesham, Christopher Bryant, Marek Rei, and Zheng Yuan. 2023. [An extended sequence tagging vocabulary for grammatical error correction](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1608–1619, Dubrovnik, Croatia. Association for Computational Linguistics.
- Detmar Meurers. 2012. Natural language processing and language learning. *Encyclopedia of applied linguistics*, pages 4193–4205.
- Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp. 2011. [Evaluating answers to reading comprehension questions in context: Results for German and the role of information structure](#). In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment*, pages 1–9, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Sarah Michaels, Mary Catherine O'Connor, Megan Williams Hall, and Lauren B Resnick. 2010. [Accountable talk@ sourcebook](#). Pittsburg, PA: Institute for Learning University of Pittsburgh. Murphy, PK, Wilkinson, IAG, Soter, AO, Hennessey, MN, & Alexander, JF.
- Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. [Towards automated document revision: Grammatical error correction, fluency edits, and beyond](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.
- Antonija Mitrovic. 2005. The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor. In *AIED*, pages 499–506.
- Antonija Mitrovic, Michael Mayo, Pramuditha Suraweera, and Brent Martin. 2001. Constraint-based tutors: a success story. In *International conference on industrial, engineering and other applications of applied intelligent systems*, pages 931–940. Springer.
- 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.
- Kaijie Mo and Renfen Hu. 2024. [ExpertEase: A multi-agent framework for grade-specific document simplification with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9080–9099, Miami, Florida, USA. Association for Computational Linguistics.
- Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghrouani, and Ossama Obeid. 2014. [The first QALB shared task on automatic text correction for Arabic](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47, Doha, Qatar. Association for Computational Linguistics.
- Ethan Mollick, Lilach Mollick, Natalie Bach, LJ Ciccarelli, Ben Przystanski, and Daniel Ravipinto. 2024. AI agents and education: Simulated practice at scale. *arXiv preprint arXiv:2407.12796*.
- Sneha Mondal, Ritika Ritika, Ashish Agrawal, Preethi Jyothi, and Aravindan Raghuvier. 2024. [DIMSIM: Distilled multilingual critics for Indic text simplification](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16093–16109, Bangkok, Thailand. Association for Computational Linguistics.

- Hyeonseok Moon, Jaewook Lee, Sugyeong Eo, Chanjun Park, Jaehyung Seo, and Heuiseok Lim. 2024. [Generative interpretation: Toward human-like evaluation for educational question-answer pair generation](#). In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2185–2196, St. Julian’s, Malta. Association for Computational Linguistics.
- Gustave Florentin Nkoulou Mvondo, Ben Niu, and Salman Eivazinezhad. 2023. Exploring The Ethical Use Of LLM Chatbots In Higher Education. *Available at SSRN 4548263*.
- Farah Nadeem and Mari Ostendorf. 2018. [Estimating linguistic complexity for science texts](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–55, New Orleans, Louisiana. Association for Computational Linguistics.
- Numaan Naeem, Kaushal Kumar Maurya, Kseniia Petukhova, and Ekaterina Kochmar. 2026a. [AITutor-EvalKit: Exploring the capabilities of AI tutors](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 457–479, Rabat, Morocco. Association for Computational Linguistics.
- Numaan Naeem, Kaushal Kumar Maurya, Kseniia Petukhova, and Ekaterina Kochmar. 2026b. [Aitutor-evalkit: Exploring the capabilities of ai tutors](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 457–479.
- Yoshinari Nagai, Teruaki Oka, and Mamoru Komachi. 2024. [A document-level text simplification dataset for Japanese](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 459–476, Torino, Italia. ELRA and ICCL.
- 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, Kentaro Inui, and Shin’ichiro Ishikawa. 2020. [Creating corpora for research in feedback comment generation](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.
- Ben Naismith, Na-Rae Han, and Alan Juffs. 2022. The university of pittsburgh english language institute corpus (PELIC). *International Journal of Learner Corpus Research*, 8(1):121–138.
- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023a. 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.
- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023b. [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.
- Hyunji Nam, Omer Gottesman, Amy Zhang, Dean Foster, Emma Brunskill, and Lyle Ungar. 2025. Efficient rl for optimizing conversation level outcomes with an llm-based tutor. *arXiv preprint arXiv:2507.16252*.
- Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. [ReadMe++: Benchmarking multilingual language models for multi-domain readability assessment](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12230–12266, Miami, Florida, USA. 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.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. [Ground truth for grammatical error correction metrics](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 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, Austin, Texas. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A fluency corpus and benchmark for grammatical error correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

- Mir Tafseer Nayeem and Davood Rafiei. 2024. KidLM: Advancing Language Models for Children—Early Insights and Future Directions. *arXiv preprint arXiv:2410.03884*.
- 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.
- 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 Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.
- Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics Conference*, pages 572–581.
- Diane Nicholls, Andrew Caines, and Paula Buttery. 2024. The Write & Improve Corpus 2024: Error-annotated and CEFR-labelled essays by learners of English.
- Allen Nie, Yash Chandak, Miroslav Suzara, Ali Malik, Juliette Woodrow, Matt Peng, Mehran Sahami, Emma Brunskill, and Chris Piech. 2025. [The GPT Surprise: Offering Large Language Model Chat in a Massive Coding Class Reduced Engagement but Increased Adopters Exam Performances](#). *Preprint*, arXiv:2407.09975.
- Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. [Exploring neural text simplification models](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.
- Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2024. [Difficult for whom? a study of Japanese lexical complexity](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 69–81, Miami, Florida, USA. Association for Computational Linguistics.
- Debora Nozza and Giuseppe Attanasio. 2023a. Is It Really That Simple? Prompting Large Language Models for Automatic Text Simplification in Italian. In *Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023)*, pages 322–333.
- Debora Nozza and Giuseppe Attanasio. 2023b. [Is it really that simple? prompting large language models for automatic text simplification in Italian](#). In *Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023)*, pages 322–333, Venice, Italy. CEUR Workshop Proceedings.
- Benjamin D. Nye, Arthur C. Graesser, and Xiaowen Hu. 2014. [AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring](#). *International Journal of Artificial Intelligence in Education*, 24(4):427–469.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashnyi. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Conference on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhashnyi, Artem Chernodub, Oleksandr Kornienko, and Igor Samokhin. 2024. [Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhashnyi. 2021. [Text Simplification by Tagging](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.
- Jenny Alexandra Ortiz-Zambrano, César Humberto Espín-Riofrío, and Arturo Montejó-Ráez. 2024. [Enhancing lexical complexity prediction through few-shot learning with gpt-3](#). In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 68–76, Torino, Italia. ELRA and ICCL.
- Robert Östling, Andre Smolentzov, Björn Tyrefors Hinnerich, and Erik Höglén. 2013. [Automated essay scoring for Swedish](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 42–47, Atlanta, Georgia. Association for Computational Linguistics.
- Leila Ouahrani and Djamel Bennouar. 2020. AR-ASAG an Arabic dataset for automatic short answer grading evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2634–2643.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- 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.
- E. B. Page. 1968. [The use of the computer in analyzing student essays](#). *International Review of Education*, 14(2):210–225.
- Ellis B Page. 1966. [The imminence of... grading essays by computer](#). *The Phi Delta Kappan*, 47(5):238–243.
- José Paladines and Jaime Ramirez. 2020. [A Systematic Literature Review of Intelligent Tutoring Systems With Dialogue in Natural Language](#). *IEEE Access*, 8:164246–164267.
- Aditya Kamlesh Parikh, Cristian Tejedor-Garcia, Catia Cucchiari, and Helmer Strik. 2026. [Rubric-guided fine-tuning of speechllms for multi-aspect, multi-rater l2 reading-speech assessment](#). *Preprint*, arXiv:2603.16889.
- Seongjin Park and Rutuja Ubale. 2023. Multitask learning model with text and speech representation for fine-grained speech scoring. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–7. IEEE.
- Jiahuan Pei, Fanghua Ye, Xin Sun, Wentao Deng, Koen Hindriks, and Junxiao Wang. 2025. [Conversational education at scale: A multi-LLM agent workflow for procedural learning and pedagogic quality assessment](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2984–2997, Suzhou, China. Association for Computational Linguistics.
- Linkai Peng, Kaiqi Fu, Binghui Lin, Dengfeng Ke, and Jinsong Zhang. 2021. [A Study on Fine-Tuning wav2vec2.0 Model for the Task of Mispronunciation Detection and Diagnosis](#). In *Proceedings of Interspeech 2021*, pages 4448–4452.
- Xian Peng, Pan Yuan, Dong Li, Junlong Cheng, Qin Fang, and Zhi Liu. 2025. Kele: A multi-agent framework for structured socratic teaching with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16342–16362.
- E. Margaret Perkoff, Abhidip Bhattacharyya, Jon Cai, and Jie Cao. 2023. [Comparing neural question generation architectures for reading comprehension](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 556–566, Toronto, Canada. Association for Computational Linguistics.
- Kseniia Petukhova and Ekaterina Kochmar. 2025. Intent Matters: Enhancing AI Tutoring with Fine-Grained Pedagogical Intent Annotation. *arXiv preprint arXiv:2506.07626*.
- Kseniia Petukhova and Ekaterina Kochmar. 2026. Towards Reward Modeling for AI Tutors in Math Mistake Remediation. *arXiv preprint arXiv:2603.24375*.
- Nhan Phan, Anna von Zansen, Maria Kautonen, Ekaterina Voskoboinik, Tamás Grósz, Raili Hilden, and Mikko Kurimo. 2024. Automated content assessment and feedback for Finnish L2 learners in a picture description speaking task. In *Proceedings of Interspeech 2024*, pages 317–321.
- Ildiko Pisan, John Lee, Chak Yan Yeung, and Jonathan Webster. 2020. [A dataset for investigating the impact of feedback on student revision outcome](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 332–339, Marseille, France. European Language Resources Association.
- Emily Pitler and Ani Nenkova. 2008. [Revisiting readability: A unified framework for predicting text quality](#). In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.
- Heather Pon-Barry, Brady Clark, Elizabeth Owen Bratt, Karl Schultz, and Stanley Peters. 2004. Evaluating the effectiveness of SCoT: A spoken conversational tutor. In *ITS 2004 Workshop on Dialog-based Intelligent Tutoring Systems*, pages 23–32.
- Mengjie Qian, Kate Knill, Stefano Banno, Siyuan Tang, Penny Karanasou, Mark J F Gales, and Diane Nicholls. 2024. Speak & Improve Challenge 2025: Tasks and Baseline Systems. *arXiv preprint arXiv:2412.11985*.
- Mengjie Qian, Rao Ma, Stefano Bannò, Kate M Knill, and Mark JF Gales. 2025a. Scaling and Prompting for Improved End-to-End Spoken Grammatical Error Correction. In *Interspeech 2025*.
- Mengjie Qian, Rao Ma, Stefano Bannò, Mark J. F. Gales, and Kate M. Knill. 2025b. [End-to-End Spoken Grammatical Error Correction](#). *Preprint*, arXiv:2506.18532.
- Xiaojuan Qian, Helen Meng, and Frank K Soong. 2012. [The use of DBN-HMMs for mispronunciation detection and diagnosis in L2 English to support computer-aided pronunciation training](#). In *Proceedings of Interspeech 2012*.
- Jipeng Qiang, Minjiang Huang, Yi Zhu, Yunhao Yuan, Chaowei Zhang, and Kui Yu. 2025. [Redefining Simplicity: Benchmarking Large Language Models from Lexical to Document Simplification](#). *Preprint*, arXiv:2502.08281.
- 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.
- Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. [Lexical Simplification with Pre-trained Encoders](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8649–8656.

- Le Qiu, Shanyue Guo, Tak-Sum Wong, Emmanuele Chersoni, John Lee, and Chu-Ren Huang. 2024. [CompLex-ZH: A new dataset for lexical complexity prediction in Mandarin and Cantonese](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 20–26, Miami, Florida, USA. Association for Computational Linguistics.
- Xin Ying Qiu and Jingshen Zhang. 2024. [Label confidence weighted learning for target-level sentence simplification](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18004–18019, Miami, Florida, USA. Association for Computational Linguistics.
- 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.
- Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. [mEditT: Multilingual text editing via instruction tuning](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 979–1001, Mexico City, Mexico. Association for Computational Linguistics.
- Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. [CoEditT: Text editing by task-specific instruction tuning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.
- Vyas Raina, Mark J. F. Gales, and Kate M. Knill. 2020. [Universal Adversarial Attacks on Spoken Language Assessment Systems](#). In *Proceedings of Interspeech 2020*, pages 3855–3859.
- Ekaterina Rakhilina, Anastasia Vyrenkova, Elmira Mustakimova, Alina Ladygina, and Ivan Smirnov. 2016. [Building a learner corpus for Russian](#). In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 66–75, Umeå, Sweden. LiU Electronic Press.
- Marek Rei and Helen Yannakoudakis. 2016. [Compositional sequence labeling models for error detection in learner writing](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.
- Jeff Rickel, Neal Lesh, Charles Rich, Candace L Sidner, and Abigail Gertner. 2002. Collaborative discourse theory as a foundation for tutorial dialogue. In *Intelligent Tutoring Systems: 6th International Conference, ITS 2002 Biarritz, France and San Sebastian, Spain, June 2–7, 2002 Proceedings 6*, pages 542–551. Springer.
- Pedro Uribe Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. [Language models and Automated Essay Scoring](#). Preprint, arXiv:1909.09482.
- Cristobal Romero and Sebastian Ventura. 2013. [Data mining in education](#). *WIREs Data Mining and Knowledge Discovery*, 3(1):12–27.
- Donya Rooein, Paul Röttger, Anastassia Shaitarova, and Dirk Hovy. 2024. [Beyond flesch-kincaid: Prompt-based metrics improve difficulty classification of educational texts](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 54–67, Mexico City, Mexico. Association for Computational Linguistics.
- Carolyn P Rosé. 2001. Interactive conceptual tutoring in Atlas-Andes. *Artif. Intell. in Education: AI-Ed in the Wired and Wireless Future*, pages 256–266.
- Ourania Rotou and André A Rupp. 2020. Evaluations of automated scoring systems in practice. *ETS Research Report Series*, 2020(1):1–18.
- Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghoulani, Ossama Obeid, and Behrang Mohit. 2015. [The second QALB shared task on automatic text correction for Arabic](#). In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35, Beijing, China. 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.
- 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).
- Vasile Rus, Sidney D’Mello, Xiangen Hu, and Arthur Graesser. 2013. Recent advances in conversational intelligent tutoring systems. *AI magazine*, 34(3):42–54.
- Michael J Ryan, Tarek Naous, and Wei Xu. 2023. [Revisiting non-English text simplification: A unified multilingual benchmark](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada. Association for Computational Linguistics.
- Horacio Saggion, Stefan Bott, Sandra Szasz, Nelson Pérez, Saúl Calderón, and Martín Solís. 2024. [Lexical complexity prediction and lexical simplification for Catalan and Spanish: Resource creation, quality assessment, and ethical considerations](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 82–94, Miami, Florida, USA. 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.
- Andreas Säuberli and Simon Clematide. 2024a. [Automatic generation and evaluation of reading comprehension test items with large language models](#). In *Proceedings of the 3rd Workshop on Tools and Resources for People with READING Difficulties (READI) @ LREC-COLING 2024*, pages 22–37, Torino, Italia. ELRA and ICCL.
- Andreas Säuberli and Simon Clematide. 2024b. Automatic Generation and Evaluation of Reading Comprehension Test Items with Large Language Models. In *Proceedings of the 3rd Workshop on Tools and Resources for People with READING Difficulties (READI)@ LREC-COLING 2024*, pages 22–37.
- Arthur Scalercio, Maria Finatto, and Aline Paes. 2024. [Enhancing sentence simplification in Portuguese: Leveraging paraphrases, context, and linguistic features](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15076–15091, Bangkok, Thailand. Association for Computational Linguistics.
- Nicy Scaria, Suma Chenna, and Deepak Subramani. 2024a. 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? In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 1–10.
- Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024b. [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?](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 1–10, Mexico City, Mexico. Association for Computational Linguistics.
- Alexander Scarlatos, Wanyong Feng, Digory Smith, Simon Woodhead, and Andrew Lan. 2024a. [Improving automated distractor generation for math multiple-choice questions with overgenerate-and-rank](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 222–231, Mexico City, Mexico. Association for Computational Linguistics.
- Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024b. Improving the validity of automatically generated feedback via reinforcement learning. In *International Conference on Artificial Intelligence in Education*, pages 280–294. Springer.
- Carolina Scarton, Gustavo Paetzold, and Lucia Specia. 2018. [SimPA: A sentence-level simplification corpus for the public administration domain](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Daniel L. Schwartz and John D. Bransford. 1998. A time for telling. *Cognition and Instruction*, 16(4):475–522.
- Laura Seiffe, Fares Kallel, Sebastian Möller, Babak Naderi, and Roland Roller. 2022. [Subjective text complexity assessment for German](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 707–714, Marseille, France. European Language Resources Association.
- Iulian Vlad Serban, Varun Gupta, Ekaterina Kochmar, Dung D. Vu, Robert Belfer, Joelle Pineau, Aaron Courville, Laurent Charlin, and Yoshua Bengio. 2020. [A Large-Scale, Open-Domain, Mixed-Interface Dialogue-Based ITS for STEM](#). *Preprint*, arXiv:2005.06616.
- Kathrin Seßler, Maurice Fürstenberg, Babette Bühler, and Enkelejda Kasneci. 2025. Can ai grade your essays? a comparative analysis of large language models and teacher ratings in multidimensional essay scoring. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 462–472.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine learning–driven language assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.
- Matthew Shardlow. 2014. [Out in the open: Finding and categorising errors in the lexical simplification pipeline](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 1583–1590, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

- Mayank Sharma, Roy Pea, and Hari Subramonyam. 2026. ConvoLearn: A Dataset of Constructivist Tutor-Student Dialogue. *arXiv preprint arXiv:2601.08950*.
- Kim Cheng Sheang, Daniel Ferrés, and Horacio Sagion. 2022. [Controllable lexical simplification for English](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 199–206, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.
- Mark D Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation: Current Applications and New Directions*. Routledge, New York, NY.
- Mark D Shermis, Jill Burstein, Derrick Higgins, and Klaus Zechner. 2010. Automated essay scoring: Writing assessment and instruction. *International encyclopedia of education*, 4(1):20–26.
- Mark D Shermis and Jill C Burstein. 2003. *Automated essay scoring: A cross-disciplinary perspective*. Routledge.
- Mark D Shermis and Joshua Wilson. 2024. *The Routledge international handbook of automated essay evaluation*. Routledge, New York, NY.
- Shamini Shetye. 2024. An evaluation of Khanmigo, a generative AI tool, as a computer-assisted language learning app. *Studies in Applied Linguistics and TESOL*, 24(1).
- Omer Shubi, Yoav Meiri, Cfir Avraham Hadar, and Yevgeni Berzak. 2024. [Fine-grained prediction of reading comprehension from eye movements](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3372–3391, Miami, Florida, USA. Association for Computational Linguistics.
- Advaith Siddharthan. 2014. [A survey of research on text simplification](#). *ITL - International Journal of Applied Linguistics*, 165(2):259–298.
- Răzvan-Alexandru Smădu, David-Gabriel Ion, Dumitru-Clementin Cercel, Florin Pop, and Mihaela-Claudia Cercel. 2024. [Investigating large language models for complex word identification in multilingual and multidomain setups](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16764–16800, Miami, Florida, USA. Association for Computational Linguistics.
- Yishen Song, Qianta Zhu, Huaibo Wang, and Qinhuo Zheng. 2024a. Automated essay scoring and revising based on open-source large language models. *IEEE Transactions on Learning Technologies*, 17:1880–1890.
- Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024b. [GEE! grammar error explanation with large language models](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.
- Shashank Sonkar, Naiming Liu, Debshila Mallick, and Richard Baraniuk. 2023. [CLASS: A design framework for building intelligent tutoring systems based on learning science principles](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1941–1961, Singapore. Association for Computational Linguistics.
- Kv Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2025. Can LLMs Reliably Simulate Real Students’ Abilities in Mathematics and Reading Comprehension? In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*, Vienna, Austria. Association for Computational Linguistics.
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM prompting strategies for joint essay scoring and feedback generation](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. 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.
- Sanja Štajner. 2021. Automatic text simplification for social good: Progress and challenges. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652.
- Sanja Štajner and Sergiu Nisioi. 2018. [A detailed evaluation of neural sequence-to-sequence models for in-domain and cross-domain text simplification](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Katherine Stasaski, Kimberly Kao, and Marti A Hearst. 2020a. CIMA: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64.
- Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020b. [CIMA: A large open access dialogue dataset for tutoring](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. [Semantic structural evaluation for text simplification](#). 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 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

- Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. [Simple and effective text simplification using semantic and neural methods](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–173, Melbourne, Australia. Association for Computational Linguistics.
- Pramuditha Suraweera and Antonija Mitrovic. 2002. KERMIT: A Constraint-Based Tutor for Database Modeling. In *Intelligent Tutoring Systems*, pages 377–387, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022. [The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4654–4662, Marseille, France. European Language Resources Association.
- Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. [Unsupervised neural text simplification](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.
- Andreas Säuberli, Diego Frassinelli, and Barbara Plank. 2025. [Do LLMs Give Psychometrically Plausible Responses in Educational Assessments?](#) *Preprint*, arXiv:2506.09796.
- 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.
- Anaïs Tack and Chris Piech. 2022. [The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues](#). *Preprint*, arXiv:2205.07540.
- Keren Tan, Kangyang Luo, Yunshi Lan, Zheng Yuan, and Jinlong Shu. 2024. [An LLM-enhanced adversarial editing system for lexical simplification](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1136–1146, Torino, Italia. ELRA and ICCL.
- LearnLM Team, Abhinot Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire, Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, and 1 others. 2024. LearnLM: Improving Gemini for learning. *arXiv preprint arXiv:2412.16429*.
- Team L, A. Wang, A. Rysbek, A. Huber, A. Nambiar, A. Kenolty, B. Caulfield, B. Lilley-Draper, B. Groot, B. Veprek, and C. Burdett. 2025. AI tutoring can safely and effectively support students: An exploratory RCT in UK classrooms. *arXiv preprint arXiv:2512.23633*.
- 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, Manchester, UK. Coling 2008 Organizing Committee.
- Amalia Todirascu, Thomas François, Delphine Bernhardt, Núria Gala, and Anne-Laure Ligozat. 2016. [Are cohesive features relevant for text readability evaluation?](#) In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 987–997, Osaka, Japan. The COLING 2016 Organizing Committee.
- Brent Townshend, Jared Bernstein, Ognjen Todic, and Eryk Warren. 1998. [Estimation of spoken language proficiency](#). In *STILL-Speech Technology in Language Learning*.
- Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv preprint arXiv:2501.06322*.
- Sean Trott and Pamela Rivière. 2024. [Measuring and modifying the readability of English texts with GPT-4](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 126–134, Miami, Florida, USA. 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.
- Masaki Uto and Yuto Uchida. 2020. Automated short-answer grading using deep neural networks and item response theory. In *International Conference on Artificial Intelligence in Education*, pages 334–339. Springer.
- 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.
- Sowmya Vajjala. 2022. Trends, Limitations and Open Challenges in Automatic Readability Assessment Research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5366–5377.
- Sowmya Vajjala. 2024. Generative Artificial Intelligence and Applied Linguistics. *JALT journal: the research journal of the Japan Association for Language Teaching*, 46(1):55–76.

- Sowmya Vajjala and Ivana Lučić. 2018. [On-eStopEnglish corpus: A new corpus for automatic readability assessment and text simplification](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.
- Sowmya Vajjala and Ivana Lucic. 2019. [On understanding the relation between expert annotations of text readability and target reader comprehension](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 349–359, Florence, Italy. Association for Computational Linguistics.
- Sowmya Vajjala and Detmar Meurers. 2012. [On improving the accuracy of readability classification using insights from second language acquisition](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.
- Sowmya Vajjala, Detmar Meurers, Alexander Eitel, and Katharina Scheiter. 2016. [Towards grounding computational linguistic approaches to readability: Modeling reader-text interaction for easy and difficult texts](#). In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 38–48, Osaka, Japan. The COLING 2016 Organizing Committee.
- Laura Vázquez-Rodríguez, Matthew Shardlow, Piotr Przybyła, and Sophia Ananiadou. 2021. [Investigating text simplification evaluation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 876–882, Online. Association for Computational Linguistics.
- Justin Vasselli, Christopher Vasselli, Adam Nohejl, and Taro Watanabe. 2023. [NAISTeacher: A prompt and rerank approach to generating teacher utterances in educational dialogues](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 772–784, Toronto, Canada. Association for Computational Linguistics.
- Mireia Vendrell and Samantha-Kaye Johnston. 2026. Scaffolding critical thinking with generative AI: Design principles for integrating large language models in higher education. *Computers and Education: Artificial Intelligence*, page 100572.
- Devika Venugopalan, Ziwen Yan, Conrad Borchers, Jionghao Lin, and Vincent Alevén. 2025. Combining large language models with tutoring system intelligence: A case study in caregiver homework support. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 373–383.
- Mabel Vogel and Carleton Washburne. 1928. An objective method of determining grade placement of children’s reading material. *The Elementary School Journal*, 28(5):373–381.
- Ekaterina Voskoboinik, Nhan Phan, Tamás Grósz, and Mikko Kurimo. 2025. Leveraging Uncertainty for Finnish L2 Speech Scoring with LLMs. In *The Workshop on Automatic Assessment of Atypical Speech*. University of Tartu Library.
- Erik Voss, Sara T Cushing, Gary J Ockey, and Xun Yan. 2023. The use of assistive technologies including generative AI by test takers in language assessment: A debate of theory and practice. *Language Assessment Quarterly*, 20(4-5):520–532.
- Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. [Sentence simplification with memory-augmented neural networks](#). 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 79–85, New Orleans, Louisiana. Association for Computational Linguistics.
- Lev S. Vygotsky. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.
- Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024a. [Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9707–9731, Bangkok, Thailand. Association for Computational Linguistics.
- Rose Wang and Dorottya Demszky. 2023. [Is ChatGPT a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 626–667, Toronto, Canada. Association for Computational Linguistics.
- Rose Wang, Pawan Wirawarn, Omar Khattab, Noah Goodman, and Dorottya Demszky. 2024b. [Back-tracing: Retrieving the cause of the query](#). In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 722–735, St. Julian’s, Malta. Association for Computational Linguistics.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024c. [Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024d. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American*

- Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199.
- Rose E Wang, Ana T Ribeiro, Carly D Robinson, Susanna Loeb, and Dora Demszky. 2024e. Tutor CoPilot: A human-AI approach for scaling real-time expertise. *arXiv preprint arXiv:2410.03017*.
- Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024f. [Large Language Models for Education: A Survey and Outlook](#). *Preprint*, arXiv:2403.18105.
- Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025a. LLM-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 510–519.
- Xinhao Wang, Keelan Evanini, Yao Qian, and Matthew Mulholland. 2021. [Automated scoring of spontaneous speech from young learners of English using transformers](#). In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 705–712.
- 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.
- Zhengxiang Wang, Veronika Makarova, Zhi Li, Jordan Kodner, and Owen Rambow. 2025b. [LLMs can perform multi-dimensional analytic writing assessments: A case study of L2 graduate-level academic English writing](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8637–8663, Vienna, Austria. Association for Computational Linguistics.
- Ziyang Wang, Sanwoo Lee, Hsiu-Yuan Huang, and Yunfang Wu. 2024g. [FPT: Feature prompt tuning for few-shot readability assessment](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 280–295, Mexico City, Mexico. Association for Computational Linguistics.
- Rebecca Watson and Ekaterina Kochmar. 2021. Read & Improve: A Novel Reading Tutoring System. In *EDM 2021: Educational Data Mining*.
- Amali Weerasinghe and Antonija Mitrovic. 2006. Individualizing self-explanation support for ill-defined tasks in constraint-based tutors. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems*, pages 56–64.
- Rodrigo Wilkens, Patrick Watrin, Rémi Cardon, Alice Pintard, Isabelle Gribomont, and Thomas François. 2024. [Exploring hybrid approaches to readability: experiments on the complementarity between linguistic features and transformers](#). In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2316–2331, St. Julian’s, Malta. Association for Computational Linguistics.
- 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.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachler. 2021. [Are We There Yet? - A Systematic Literature Review on Chatbots in Education](#). *Frontiers in Artificial Intelligence*, Volume 4 - 2021.
- Kristian Woodsend and Mirella Lapata. 2011. [Learning to simplify sentences with quasi-synchronous grammar and integer programming](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Beverly Park Woolf. 2010. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.
- Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023a. [ChatGPT or Grammarly? Evaluating ChatGPT on Grammatical Error Correction Benchmark](#). *Preprint*, arXiv:2303.13648.
- M. Wu, K. Li, W.-K. Leung, and H. Meng. 2021a. [Transformer Based End-to-End Mispronunciation Detection and Diagnosis](#). In *Proceedings of Interspeech 2021*, pages 3954–3958.
- Minglin Wu, Kun Li, Wai-Kim Leung, and Helen Meng. 2021b. [Transformer Based End-to-End Mispronunciation Detection and Diagnosis](#). In *Proceedings of Interspeech 2021*, pages 3954–3958.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2023b. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*.
- Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. [Sentence simplification by monolingual machine translation](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.
- Xiaoming Xi. 2023. Advancing language assessment with AI and ML—Leaning into AI is inevitable, but can theory keep up? *Language Assessment Quarterly*, 20(4-5):357–376.

- Xiaoming Xi, Derrick Higgins, Klaus Zechner, and David M. Williamson. 2008. [Automated scoring of spontaneous speech using SpeechRaterSM v1.0](#). Technical Report 2, ETS Research Report Series.
- Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. [Text readability assessment for second language learners](#). In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.
- Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023a. [Evaluating reading comprehension exercises generated by LLMs: A showcase of ChatGPT in education applications](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 610–625, Toronto, Canada. Association for Computational Linguistics.
- Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023b. Evaluating reading comprehension exercises generated by LLMs: A showcase of ChatGPT in education applications. In *Proceedings of the 18th workshop on innovative use of NLP for building educational applications (BEA 2023)*, pages 610–625.
- Bin Xu, Yu Bai, Huashan Sun, Yiguan Lin, Siming Liu, Xinyue Liang, Yaolin Li, Yang Gao, and Heyan Huang. 2025. EduBench: A Comprehensive Benchmarking Dataset for Evaluating Large Language Models in Diverse Educational Scenarios. *arXiv preprint arXiv:2505.16160*.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. [Problems in current text simplification research: New data can help](#). *Transactions of the Association for Computational Linguistics*, 3:283–297.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. [Optimizing statistical machine translation for text simplification](#). *Transactions of the Association for Computational Linguistics*, 4:401–415.
- Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, and Long Ma. 2021. [Explore wav2vec 2.0 for Mispronunciation Detection](#). In *Proceedings of Interspeech 2021*, pages 4428–4432.
- Taichi Yamashita. 2025. [Exploring potential biases in GPT-4o’s ratings of English language learners’ essays](#). *Language Testing*, 42(3):344–358.
- Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2024. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, 55(1):90–112.
- Kevin P Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023a. 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.
- Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023b. [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.
- Diya Yang, Sherry Tongshuang Wu, and Marti A. Hearst. 2024. [Human-AI interaction in the age of LLMs](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, pages 34–38, Mexico City, Mexico. 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.
- Helen Yannakoudakis and Ronan Cummins. 2015. [Evaluating the performance of automated text scoring systems](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–223, Denver, Colorado. Association for Computational Linguistics.
- Jing Yao, Xiaoyuan Yi, Xiting Wang, Jindong Wang, and Xing Xie. 2023. From Instructions to Intrinsic Human Values—A Survey of Alignment Goals for Big Models. *arXiv preprint arXiv:2308.12014*.
- Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, and 1 others. 2025. EX-CGEC: A Benchmark for Edit-Wise Explainable Chinese Grammatical Error Correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 25678–25686.
- Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. 2025. [DREsS: Dataset for rubric-based essay scoring on EFL writing](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13439–13454, Vienna, Austria. Association for Computational Linguistics.

- Su-Youn Yoon, Lisa Pierce, Amanda Huensch, Eric Juul, Samantha Perkins, Richard Sproat, and Mark Hasegawa-Johnson. 2009. Construction of a rated speech corpus of L2 learners' spontaneous speech. *CALICO Journal*, 26(3):662–673.
- Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. [Neural and FST-based approaches to grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy. Association for Computational Linguistics.
- Anđelka Zečević, Milica Ćulafić, and Stefan Stojković. 2024. [On simplification of discharge summaries in Serbian: Facing the challenges](#). In *Proceedings of the First Workshop on Patient-Oriented Language Processing (CL4Health) @ LREC-COLING 2024*, pages 104–108, Torino, Italia. ELRA and ICCL.
- Klaus Zechner and Keelan Evanini. 2019. *Automated speaking assessment: Using language technologies to score spontaneous speech*. Routledge.
- Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. [Task-independent features for automated essay grading](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–232, Denver, Colorado. Association for Computational Linguistics.
- Tatsuya Zetsu, Yuki Arase, and Tomoyuki Kajiwar. 2024. [Edit-constrained decoding for sentence simplification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7161–7173, Miami, Florida, USA. Association for Computational Linguistics.
- Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yukai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. *speechocean762: An Open-Source Non-Native English Speech Corpus for Pronunciation Assessment*. In *Proceedings of Interspeech 2021*, pages 3710–3714.
- Xingxing Zhang and Mirella Lapata. 2017. [Sentence simplification with deep reinforcement learning](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.
- Guoguo Zhao, Sinem Sonsaat, Apimee Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. 2018a. [L2-ARCTIC: A Non-native English Speech Corpus](#). In *Proceedings of Interspeech 2018*, pages 2783–2787.
- Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018b. [Integrating transformer and paraphrase rules for sentence simplification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.
- Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. [Improving Seq2Seq grammatical error correction via decoding interventions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.
- Zhemini Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. [A monolingual tree-based translation model for sentence simplification](#). In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.
- Ramon Ziai, Niels Ott, and Detmar Meurers. 2012. Short answer assessment: Establishing links between research strands. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 190–200.

A Additional References

We provide more detailed references for additional reading on specific topics in this section, grouping them along the four dimensions: writing, speaking, reading, and tutoring, in the Tables 1–4.

Written GEC & GED	
Surveys	Bryant et al. (2023)
Datasets	Yannakoudakis et al. (2011); Dahlmeier et al. (2013); Dale et al. (2012); Ng et al. (2013, 2014); Mohit et al. (2014); Rozovskaya et al. (2015); Napoles et al. (2017); Bryant et al. (2019); Rozovskaya and Roth (2019); Koyama et al. (2020); Náplava et al. (2022); Masciolini et al. (2025)
Evaluation	Dahlmeier and Ng (2012); Felice and Briscoe (2015); Napoles et al. (2015, 2016); Bryant et al. (2017); Choshen et al. (2020); Belkebir and Habash (2021)
Pre-LLM Approaches	Chodorow et al. (2007); Kochmar et al. (2012); Felice et al. (2014); Junczys-Dowmunt and Grundkiewicz (2014, 2016); Junczys-Dowmunt et al. (2018); Yuan et al. (2019); Malmi et al. (2019); Stahlberg and Kumar (2020); Kaneko et al. (2020); Omelianchuk et al. (2020); Mallinson et al. (2020); Katsumata and Komachi (2020); Mallinson et al. (2022); Alhafni et al. (2023); Zhou et al. (2023); Mesham et al. (2023); Alhafni and Habash (2025)
LLM Approaches	Fang et al. (2023); Coyne et al. (2023); Wu et al. (2023a); Loem et al. (2023); Raheja et al. (2023); Davis et al. (2024); Katinskaia and Yangerber (2024); Omelianchuk et al. (2024); Kaneko and Okazaki (2023); Katinskaia and Yangerber (2024); Raheja et al. (2024); Omelianchuk et al. (2024); Kobayashi et al. (2024); Mita et al. (2024)
GEE	
Datasets	Nagata (2019); Nagata et al. (2020); Pilan et al. (2020); López Cortez et al. (2024); Kobayashi et al. (2024); Coyne et al. (2025)
Pre-LLM Approaches	Nagata (2019); Pilan et al. (2020)
LLM Approaches	López Cortez et al. (2024); Kobayashi et al. (2024)
Automatic Writing Assessment	
Surveys	Shermis and Burstein (2003); Shermis et al. (2010); Shermis and Burstein (2013); Ke and Ng (2019); Beigman Klebanov and Madnani (2020, 2022); Li and Ng (2024b,a); Shermis and Wilson (2024)
Datasets	Granger et al. (1993); Yannakoudakis et al. (2011); Blanchard et al. (2013); Geertzen et al. (2013); Ishikawa (2013); Östling et al. (2013); Boyd et al. (2014); Rakhilina et al. (2016); Horbach et al. (2017); Mathias and Bhattacharyya (2018); Glaznieks et al. (2023); Marinho et al. (2021); Habash and Palfreyman (2022); Naismith et al. (2022); Crossley et al. (2023); Nicholls et al. (2024); Imperial et al. (2025); Yoo et al. (2025); Wang et al. (2025b); Bashendy et al. (2026)
Evaluation	Williamson et al. (2012); Yannakoudakis and Cummins (2015); Buzick et al. (2016); Rotou and Rupp (2020)
Pre-LLM Approaches	Burstein (2002); Landauer et al. (2002); Rudner et al. (2006); Yannakoudakis et al. (2011); Chen and He (2013); Zesch et al. (2015); Alikaniotis et al. (2016); Vajjala (2018); Rodriguez et al. (2019); Yang et al. (2020); Wang et al. (2022)
LLM Approaches	Mizumoto and Eguchi (2023); Yancey et al. (2023b); Banno et al. (2024); Song et al. (2024a); Stahl et al. (2024); Atkinson and Palma (2025); Cai et al. (2025); Yamashita (2025); Seßler et al. (2025); Chu et al. (2025a)
Short Answer Scoring	
Surveys	Ziai et al. (2012); Burrows et al. (2015)
Datasets	Meurers et al. (2011); Ouahrani and Bennouar (2020)
Pre-LLM Approaches	Leacock and Chodorow (2003); Uto and Uchida (2020); Horbach et al. (2024)
LLM Approaches	Chamieh et al. (2024); Ferreira Mello et al. (2025)

Table 1: Additional references for writing tasks (assistive/assessment)

Readability Assessment	
Surveys	Collins-Thompson (2014) ; Vajjala (2022)
Datasets	Paetzold and Specia (2016) ; Vajjala and Lučić (2018) ; Shardlow et al. (2021) ; Seiffe et al. (2022) ; Naous et al. (2024)
Evaluation	Vajjala et al. (2016) ; Todorascu et al. (2016) ; Vajjala and Lucic (2019) ; Shubi et al. (2024)
Pre-LLM Approaches	Collins-Thompson and Callan (2004) ; Pitler and Nenkova (2008) ; Feng et al. (2010) ; Vajjala and Meurers (2012) ; Xia et al. (2016) ; Nadeem and Ostendorf (2018) ; Azpiazu and Pera (2019) ; Deutsch et al. (2020) ; Lee et al. (2021) ; Wilkens et al. (2024)
LLM Approaches	Lee and Lee (2023) ; Nohejl et al. (2024) ; Rooein et al. (2024) ; Wang et al. (2024g) ; Smădu et al. (2024)
Text Simplification	
Surveys	Siddharthan (2014) ; Alva-Manchego et al. (2020)
Datasets	Zhu et al. (2010) ; Coster and Kauchak (2011) ; Kauchak (2013) ; Hwang et al. (2015) ; Xu et al. (2015, 2016) ; Kajiwarra and Komachi (2016) ; Zhang and Lapata (2017) ; Sulem et al. (2018b) ; Scarton et al. (2018) ; Vajjala and Lučić (2018) ; Saggion et al. (2022) ; Hayakawa et al. (2022) ; Ryan et al. (2023) ; Alhafni et al. (2024) ; Shardlow et al. (2024) ; Jiang and Xu (2024) ; Saggion et al. (2024) ; Qiu et al. (2024) ; Nagai et al. (2024)
Evaluation	Xu et al. (2016) ; Sulem et al. (2018a) ; Vásquez-Rodríguez et al. (2021) ; Alva-Manchego et al. (2021) ; Cardon et al. (2022) ; Huang and Kochmar (2024) ; Agrawal and Carpuat (2024)
Pre-LLM Approaches	Chandrasekar et al. (1996) ; Elhadad and Sutaria (2007) ; Zhu et al. (2010) ; Woodsend and Lapata (2011) ; Wubben et al. (2012) ; Kajiwarra et al. (2013) ; Shardlow (2014) ; Xu et al. (2016) ; Paetzold and Specia (2016) ; Nisioi et al. (2017) ; Zhang and Lapata (2017) ; Alva-Manchego et al. (2017) ; De Hertog and Tack (2018) ; Štajner and Nisioi (2018) ; Guo et al. (2018) ; Maddela and Xu (2018) ; Zhao et al. (2018b) ; Gooding and Kochmar (2018) ; Vu et al. (2018) ; Gooding and Kochmar (2019) ; Surya et al. (2019) ; Qiang et al. (2020) ; Martin et al. (2020) ; Omelianchuk et al. (2021) ; Maddela et al. (2021) ; Qiang et al. (2021) ; Hazim et al. (2022) ; Martin et al. (2022) ; Sheang et al. (2022)
LLM Approaches	Chi et al. (2023) ; Nozza and Attanasio (2023b) ; Kew et al. (2023b) ; Trott and Rivière (2024) ; Scalercio et al. (2024) ; Mondal et al. (2024) ; Zečević et al. (2024) ; Tan et al. (2024) ; Hewett et al. (2024) ; Qiu and Zhang (2024) ; Zetsu et al. (2024) ; Asthana et al. (2024) ; Farajidizaji et al. (2024) ; Mo and Hu (2024) ; Barayan et al. (2025)
Question Generation	
Surveys	Kurdi et al. (2020) ; Al Faraby et al. (2024a) ; Flor (2025)
Datasets	Chen et al. (2018a)
Evaluation	Horbach et al. (2020) ; Xiao et al. (2023b) ; Gorgun and Bulut (2024) ; Deroy et al. (2025)
Pre-LLM Approaches	Flor (2025, ch4–ch9)
LLM Approaches	Flor (2025, ch10) , Al Faraby et al. (2024b) ; Säuberli and Clematide (2024b) ; Scaria et al. (2024a) ; Kumar and Lan (2024)

Table 2: Additional references for reading tasks (assistive/assessment)

Spoken GEC & GED	
Evaluation	Lu et al. (2022); Qian et al. (2025b)
Datasets	Izumi et al. (2004); Caines et al. (2016); Kim et al. (2024a); Knill et al. (2024)
Pre-LLM Approaches	Izumi et al. (2003); Lee et al. (2011); Knill et al. (2019); Lu et al. (2019b,a, 2020, 2022); Bannò et al. (2023b, 2024a); Karanasou et al. (2025); Qian et al. (2025a,b)
LLM Approaches	Lu et al. (2025)
Spoken Language Assessment	
Surveys	Zechner and Evanini (2019)
Datasets	Menzel et al. (2000); Izumi et al. (2004); Yoon et al. (2009); Ishikawa (2014); Baur et al. (2017, 2018); Zhao et al. (2018a); Baur et al. (2019); Ishikawa (2019); Zhang et al. (2021); Coulange et al. (2024); Kim et al. (2024a); Knill et al. (2024)
Pre-LLM Approaches	Bernstein et al. (1990); Cucchiari et al. (1997); Townshend et al. (1998); Franco et al. (2000); Xi et al. (2008); Qian et al. (2012); Malinin et al. (2017); Chen et al. (2018b); Evanini et al. (2018); Craighead et al. (2020); Raina et al. (2020); Peng et al. (2021); Wu et al. (2021a); Xu et al. (2021); Wang et al. (2021); Kim et al. (2022); Bannò et al. (2023a); McKnight et al. (2023); Park and Ubale (2023)
LLM Approaches	Fu et al. (2024); Phan et al. (2024); Bannò et al. (2025b); Ma et al. (2025); Voskoboinik et al. (2025)

Table 3: Additional references for speaking tasks (assistive/assessment)

Intelligent Tutoring Systems

Surveys	Paladines and Ramirez (2020) ; Wollny et al. (2021) ; Wang et al. (2024f)
Datasets	Michaels et al. (2010) ; Caines et al. (2020) ; Stasaski et al. (2020b) ; Suresh et al. (2022) ; Demszky and Hill (2023) ; Macina et al. (2023a) ; Wang et al. (2024d) ; Maurya et al. (2025) ; Sharma et al. (2026)
Evaluation	Demszky et al. (2021) ; Vasselli et al. (2023) ; Jurenka et al. (2024) ; Kochmar et al. (2025) ; Maurya et al. (2025) ; Hardy and Kim (2026) ; Naeem et al. (2026b)
Pre-LLM Approaches	Evers and Nijholt (2000) ; Freedman (2000) ; Mitrovic et al. (2001) ; Rosé (2001) ; Suraweera and Mitrovic (2002) ; Anderson et al. (2004) ; Graesser et al. (2004, 2006) ; Weerasinghe and Mitrovic (2006) ; Dzikovska et al. (2010) ; D’Mello et al. (2012) ; Romero and Ventura (2013) ; Nye et al. (2014) ; Serban et al. (2020) ; Watson and Kochmar (2021)
LLM Approaches	Tack and Piech (2022) ; Elkins et al. (2023) ; Lee et al. (2023) ; Nay-eem and Rafiei (2024) ; Markel et al. (2023) ; Sonkar et al. (2023) ; Tack et al. (2023) ; Vasselli et al. (2023) ; Wang and Demszky (2023) ; Chowdhury et al. (2024) ; Daheim et al. (2024) ; Denny et al. (2024) ; Elkins et al. (2024) ; Liu et al. (2024a) ; Scarlato et al. (2024b) ; Wang et al. (2024a,b,c) ; Nam et al. (2025) ; Nie et al. (2025) ; Peng et al. (2025) ; Srivatsa et al. (2025) ; Petukhova and Kochmar (2025, 2026)

Table 4: Additional references for tutoring tasks (assistive/assessment)

Fine-Grained Content Zone Prediction in German Argumentative Essays Using LLMs

Xiaoyu Bai and Manfred Stede

Applied Computational Linguistics

University of Potsdam

Karl-Liebknecht-Straße 24-25

14476, Potsdam, Germany

{xiaoyu.bai | stede}@uni-potsdam.de

Abstract

We introduce FDE-Arg, a newly compiled dataset of argumentative student essays in German. We use two Llama models of different sizes to label sentence-level content zones both in FDE-Arg and in an existing dataset of source-dependent argumentative essays. We investigate three approaches for improving model performance: a) Incorporating targeted task information into the prompt text; b) few-shot prompting with up to 10 examples selected on the basis of similarity with the target instance; and c) parameter-efficient fine-tuning. We observe that both incorporating additional information in the prompts and similarity-based few-shot prompting have produced highly promising performance gains over the baseline.

1 Introduction

Given that written argumentation is a central skill set taught in secondary schools, using the latest NLP applications to support the teaching and analysis of argumentation in student writing is highly desirable. One central subtask is automatically recognising what role individual text units play with respect to the overall text function and extracting those that are relevant to argumentation. As such, the task is related to argumentative zoning as established by Teufel et al. (1999), which targets a different text genre, viz. scientific writing, and analyses argumentative and rhetorical structure in academic papers. Applied to the educational domain, such an analysis of the functional components of a student’s argumentative text can contribute to a better understanding of the student’s argumentation strategy and form the basis of constructive feedback.

In our present work, we investigate the application of LLMs to this task, which we refer to as “(argumentative) content zone prediction” in German-language argumentative essays. The aim is to automatically assign one out of a set of content

zone labels to each sentence in the essay, where the label reveals the functional and argumentative role played by the sentence w.r.t. the essay. To illustrate, Figure 1 shows a fictitious example in English that is loosely adapted from an authentic essay from our German data.

At our school, there is currently an ongoing discussion on “healthy diet”. The question is: “What is better? Fast food or cooking oneself?”
One position is that cooking at home is better than fast food. That’s much healthier. It’s also a much better option when one is expecting guests, who will for sure prefer something homemade with love over something that’s just store-bought and mixed together. Cooking together with loved ones can also make for great memories.
A strong statement made by another student is this: I’d rather order a pizza than spend hours at the stove. A main argument for this position is convenience. Cooking oneself takes much more effort than just getting a pizza. We have to understand that not everyone has the energy and the motivation to cook after a long day’s work. Also, ordering fast food can save you a lot of time.
I personally think that it can be fun to cook, but it’s also time-consuming. So my take on this is this: One should generally cook at home, but it’s also ok to order a pizza from time to time.

Argumentative content zone labels

Non-argumentative: Segments that do not introduce new argumentative elements.

Thesis 1: Claim w.r.t. the topic that aligns with the writer’s stance.

Pro-argument: Premise that supports Thesis 1.

Thesis 2: Claim that aligns with the counter-position.

Con-argument: Premise that supports Thesis 2.

Central thesis: Writer’s core position on the topic.

Figure 1: Fictitious example essay with colour-coded annotations, loosely adapted from an authentic German essay from our data.

For this, we use two datasets of argumentative essays in German: The recently released GerTE dataset (Bai and Stede, 2025) featuring source-based essays that are written with reference to reading material, and our novel dataset FDE-Arg, featuring stand-alone essays. Each dataset uses its own set of content zone labels that reflect the respective writing task. For instance, labels in GerTE differentiate between arguments repeated from the reading material and original arguments by the writer (Bai and Stede, 2025), while labels in FDE-Arg differentiate between the writer’s position on a discussion topic and the counter-position that the writer concedes.

Bai and Stede (2025) have conducted baseline zone prediction experiments on GerTE using a small Llama model. We take their approach as

our starting point and investigate ways to improve LLM performance on the argumentative content zone prediction task, using both GerTE and FDE-Arg. Concretely, we experiment with the following three strategies:

- Enhancing the prompt text with task-specific background information
- Few-shot prompting with selection of example instances based on semantic similarity
- Parameter-efficient fine-tuning (PEFT) with QLoRA (Dettmers et al., 2023)

Our overall contributions are two-fold: First, We highlight a novel dataset of German argumentative essays, annotated with a fine-grained set of content zone labels. Second, our experiments reveal that both the inclusion of targeted information in the prompt and few-shot prompting are highly promising directions.

Our code and our resources for FDE-Arg, including the annotation guide as well as annotated datasets, are made publicly available.¹

2 Related Work

Scoring and analysing argumentative essays by students of different age groups is a well-established area of interest in educational NLP: Argumentative essays are featured in well-known datasets such as ASAP-AES² and ETS TOEFL11 (Blanchard et al., 2013), which have been widely used in automated essay scoring research (Bai and Stede, 2023). Moreover, the Argument Annotated Essays Corpus (Stab and Gurevych, 2017) and the PERSUADE Corpus (Crossley et al., 2022, 2024) are two prominent examples of English-language learner/student essays which have been annotated with argument mining (AM) information, such as claims and premises, and which therefore allow for the computational analysis of students’ argumentation strategies.

With respect to German, DARIUS (Schaller et al., 2024b) is a 4,500-sample corpus of argumentative essays that are written by secondary school students and labelled with multiple layers of argumentation-related annotations, including coarse-grained content zones and major claims.

Schaller et al. (2024a) have investigated automatically predicting the labels in DARIUS using both supervised models and an LLM, although their focus is on evaluating the fairness of different models while performing prediction. Stahl et al. (2024) provide a comparable, 1320-sample dataset of student essays with AM labels on multiple levels of granularity. They have also conducted baseline label prediction experiments using a BERT-based model. On the level of university education, Wambsganss et al. (2020a,b) have compiled a 1000-sample corpus of persuasive essays by business administration students and have used it to develop a tool for analysing the argumentation structure in the essays and providing feedback on them.

While to our knowledge LLMs have not yet been extensively applied to these datasets, they have been increasingly applied to the field of AM: Recent work has looked into using LLMs in different prompt scenarios to perform canonical subtasks of AM, such as argument component type recognition and classification (Chen et al., 2024) and argument relation extraction (Gorur et al., 2025). Fine-tuning smaller LLMs using PEFT have also shown to be successful for similar core AM tasks (Kawarada et al., 2024; Cabessa et al., 2025).

3 Data

We study argumentative content zone prediction with LLMs, applied to two German corpora with source-dependent and stand-alone essays.

3.1 GerTE

GerTE (Bai and Stede, 2025) is a corpus consisting of 117 essays on three topics. The essays have been collected in the context of a source-based writing exercise where the writer first reads a news article on one of three possible topics, then writes an essay that presents the topic, discusses the arguments put forward in the source article and finally concludes with a stance of their own. While the writing task is a common exercise in German secondary schools, GerTE was not collected from authentic school students due to legal concerns. Instead, the authors recruited crowd workers who took on the role of students and composed the essays.

Essays in GerTE have a mean length of approximately 270 words. After automatic sentence-segmentation, the corpus is annotated on the sentence level with functional content zones, where each zone describes a sentence’s functional role to

¹<https://github.com/discourse-lab/FairDebArgMining>

²<https://www.kaggle.com/c/asap-aes>

the essay. GerTE uses an inventory consisting of the following five class labels:

- **info_intro**: Introductory sentences that present the topic and/or give publication-related information on the source article.
- **article_pro**: Sentences denoting pro-arguments from the source article
- **article_con**: Sentences denoting contra-arguments from the source article
- **own**: Sentences denoting the writer’s own position and arguments
- **other**: Sentences that do not fit into the above four classes

The full corpus consists of 1713 sentences from the 117 essays, with each sentence being labelled with exactly one of the five content zones. Label distribution is unbalanced, with “own” being the most frequent and “other” being the least frequent class. The exact distribution is provided in Table 1.

Label	own	a_pro	info_intro	a_con	other
Count	551	460	281	243	178

Table 1: Class-specific label counts in GerTE

3.2 FDE-Arg

Our new dataset comprises stand-alone essays written by students in the 9th grade of various German secondary schools. It was originally collected by education scientists in the context of the *Fair Debating and Written Argumentation* (FDE) project, which aims to investigate and improve the written argumentation skills of 9th-grade students in Germany (Giera et al., 2025a,b). The students were asked to compose pro-and-contra argumentations that should address both sides of a discussion topic, weigh the arguments and draw a conclusion with their own position. Each essay deals with one of four possible, easily accessible topics such as “Is fast food better than cooking at home?” or “Should one undertake volunteer work?”. The full FDE dataset comprises 1,061 essays. On this basis, Bai et al. (2026) have developed a scheme for annotating the essays with argumentative content zones³, which they have applied to and validated on 50 samples. Adopting their scheme, we have annotated

³However, they refer to them as *argument component types*, following argumentation mining terminology.

another 50 essays, which yields a total set of 100 essays labelled with argumentative content zones. We refer to this 100-sample dataset as *FDE-Arg*.

On the essay level, each text is assigned an “overall argumentative constellation”, with “**decided**” denoting that the author argues in favour of a particular stance in the discussion, with or without discussing the opposing stance; “**undecided**” denotes that the author describes both sides of the discussion and refrains from taking a position of their own. Moreover, the essays have been manually segmented into sentences or phrases that form argumentative units,⁴ and each segment is labelled with one of the following six labels that describe the segment’s argumentative function in the essay:

- **central thesis (cth)**: Segment describing the author’s central stance on the topic. If the author subscribes to one side of the discussion without addressing the counter-position, this could be the only thesis/claim in the essay. In essays with the “undecided” overall constellation, “cth” can also be an explicit statement that the author cannot make up their mind.
- **thesis 1 (th1) / thesis 2 (th2)**: Segments describing the two opposing positions on the discussion topic if both are present. In an essay with the “decided” constellation, where the author, however, also acknowledges the counter-position, “th1” denotes the position of the author and is therefore in line with the author’s central stance (“cth”), whereas “th2” denotes the counter-position.
- **pro argument (pro) / con argument (con)**: Segment describing arguments that support the theses. Those supporting “th1” and/or “cth” are denoted as “pro arguments” and those supporting “th2” as “con arguments”.
- **non argumentative (n-a)**: Segment that is not considered argumentative. This includes sentences that serve to structure the essay or that present background information and should be given to all segments not covered by the above labels.

⁴Their rule for manual segmentation is that complex sentences are split into smaller segments if they individually play a role to the argumentative structure of the essay. Those that are not relevant to argumentation, e.g. because they only give background information or serve to structure the essay, are left as they are. We refer to Bai et al. (2026) and their annotation guide for details.

We refer to Bai et al. (2026) for details on the annotation process for FDE-Arg, including theoretical and practical motivations for the scheme used. Moreover, they also present an inter-annotator agreement study based on 30 essays, which has achieved an overall Cohen’s κ of 0.79.

On average, essays in FDE-Arg are longer than those in GerTE, with a mean word count of 345 per essay. The 100 essays comprise 2,595 labelled segments in total; the label distribution is shown in Table 2. It is unsurprising that there are far fewer segments labelled as “cth”, “th1” and “th2” than segments labelled as “pro” and “con” since a single position or claim can be supported by multiple arguments.

Label	pro	con	n-a	cth	th1	th2
Count	911	697	688	106	97	96

Table 2: Class-specific label counts in FDE-Arg

To summarise, Table 3 captures and contrasts central characteristics of GerTE and FDE-Arg.

4 Methods

We use two open-source, instruction fine-tuned Llama 3 models of different sizes: The smaller “Llama-3.1-8B-Instruct”⁵ with 8 billion parameters (hereafter referred to as “Llama-8B”) and the larger “Llama-3.3-70B-Instruct”⁶ with 70 billion parameters (hereafter “Llama-70B”). While newer, more powerful LLMs are a potential way to achieve better prediction results, our focus is on different strategies to enhance model performance while keeping the LLM models themselves fixed. Each of our three strategies is described in detail in the sections to come. Our experiments used the Huggingface framework (Wolf et al., 2020) and Groq API⁷.

4.1 Baselines

For GerTE, Bai and Stede (2025) have conducted first LLM-based content zone prediction experiments using Llama-8B. They experimented with a zero-shot prompt setting, with and without inclusion of the source article text in the prompt, as well as a one-shot setting using a manually selected example essay for in-context learning. They report

⁵<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

⁶<https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>

⁷<https://groq.com/>

their best result as a weighted-average F1 of **0.661**. In contrast, their best-performing ensemble model using BERT and SVM achieved an F1 of 0.713. We treat the prompting approach by Bai and Stede (2025) as our baseline setting on GerTE.

Bai et al. (2026) have also conducted a first set of LLM-experiments, albeit on the initial 50-sample portion of FDE-Arg which they had annotated. Their base prompt text is similar to that by Bai and Stede (2025): It asks the LLM to adopt the role of a 9th-grade teacher, explains the meaning of each label, supplies the segmented essay and asks the model to output a sequence of labels of the same length. Instead of the three-letter labels such as “cth” or “pro” in the dataset, for the prompt text, the zone labels are mapped to interpretable counterparts such as “Zentrale_These” (“central thesis”) and “Pro_Argument” (“pro argument”), respectively. The exact German-language prompt text is provided in the Appendix. Using both Llama-8B and Llama-70B, Bai et al. (2026) evaluated zero-shot prompting and two-shot prompting with two randomly chosen example essays, one with a “decided” and one with an “undecided” overall constellation. We adopt these two prompt settings as our baseline on FDE-Arg.

4.2 Additional Information in Prompt Text

As the most simple strategy, we experimented with enhancing the baseline prompt for either dataset with additional information that addresses specific challenges in the respective dataset.

4.2.1 GerTE: Summary of Source Text Content

The baseline by Bai and Stede (2025) has already explored incorporating the source article text into the LLM prompt. This has indeed benefitted the LLM’s prediction, although the resulting performance was still far below that of supervised models. According to the authors’ error analysis, a significant source of errors is confusion between pro and con arguments from the source article as well as their distinction from the writer’s own arguments. Since a solid understanding of the main arguments from the source text is essential to avoiding such confusion, we manually extracted and summarised the main arguments from each source article in concise sentences⁸ and included them in the system

⁸These summarised main arguments are in fact what has been referred to as “reference snippets” in Bai and Stede (2022). In a preparatory step to composing the essays in

	GerTE	FDE-Arg
Essay type	source-dependent	stand-alone
Mean word count per essay	270	345
Label set	own, article_pro, article_con, info_intro, other	cth, th1, th2, pro, con, n-a
Annotation unit	sentence (automatic sentence-segmentation)	argumentative segment (manual segmentation)
Total number of essays	117	100
Total number of annotated units	1,713	2,595

Table 3: Central characteristics of GerTE and FDE-Arg

prompt alongside the source article itself. A gloss of our addition is thus: *The following are the arguments that are brought up in the text. The pro arguments are these [LIST OF PRO-ARGUMENTS]. The con arguments are these [LIST OF CONTRA-ARGUMENTS].* The full system prompt for an essay on the topic of using social media in class is provided in the Appendix, including our formulation of the article’s main arguments.

4.2.2 FDE-Arg: Instructions from Annotation Guide

The annotation guide for FDE-Arg provides detailed information and annotation strategies on the dataset. Therefore, we aimed to improve LLM performance by adding relevant content from the annotation guide to the prompt text, experimenting with two sets of information in particular:

First, various authors of the essays conclude their essay by stating or repeating their overall view on the discussion topic. According to the annotation scheme, such a statement should be annotated as “central thesis (cth)” only if no previous segment has been annotated as such. Otherwise, it is considered a conclusion and thus an essay structuring element that is annotated as “non-argumentative (n-a)”. To reduce confusion between “cth” and “n-a”, we added a formulation of this annotation rule to the prompt. The exact formulation of this addition is given in the Appendix.

Second, for further analysis of an essay’s argumentation structure, it is particularly crucial that the theses, particularly the central thesis, are correctly extracted, and human annotators are instructed by the annotation guide to prioritise the labels for theses over those for arguments in cases of doubt. To encourage the same behaviour from LLMs and to target better prediction of the thesis labels, we added a short instruction based on the

recommended annotation steps from the annotation guide. A gloss of our addition is as follows: *When assigning zone labels to segments you can proceed as follows: 1. Find the central message of the essay (“cth”) ... 2. Find further theses (“th1” / “th2”) ... 3. Find the arguments (“pro” / “con”) ... 4. Mark the remaining, unassigned segments (“n-a”)* We refer to the Appendix again for the full addition.

4.3 Similarity-Based Few-Shot Prompting

In contrast to the one-shot and two-shot prompts in the baseline settings (see above), we experimented with few-shot prompting using up to 10 example essays. Moreover, previous work on example selection for in-context learning have shown that example instances in few-shot prompting are especially effective when they display high levels of similarity with the target instance (Liu et al., 2022; Ajjour and Wachsmuth, 2025). We therefore adopted a similarity-based method to individually and automatically select a set of example essays for each target essay, as described in the following.

Although we did not *train* the models in any sense, we divided our data into a training and a test set, of which the training set served as the pool of essays from which examples could be selected. For a given test essay t at inference time, we first extracted all training essays that address the same topic as t , treating them as “candidate examples” C . Next, we ranked all essays in C based on their semantic similarity with t and sorted them in descending order. From this ranked list of candidate essays, we then selected the top k essays, where k is the number of examples we used in the few-shot prompting scheme.

Semantic similarity between a given candidate essay $c \in C$ and t was computed using SBERT (Reimers and Gurevych, 2019), specifically the pre-trained model “sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2”⁹, and cosine simi-

GerTE, writers were in fact asked to write out the main arguments from the source article. These summarised arguments we include in the prompts have originally been manually compiled to serve as reference answers for this preparatory task.

⁹<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>

larity. For this calculation, we split both c and t into three chunks, respecting sentence boundaries, and recorded the similarity between the index-aligned pairs of chunks. This produced three similarity values for each essay pair. The final similarity value between the essay pair was computed as a weighted average of the three chunk-level similarity scores. Our similarity-based example ranking process is illustrated in Figure 2.

The motivation for splitting each candidate and target essay into three chunks is two-fold: First, entire essays would exceed the 128-token sequence length limit of the SBERT model used, which would subject them to truncation. Second, by taking a weighted average of chunk-level similarities, we could control the weight put on the similarity in specific areas of essays. For FDE-Arg essays, we assigned equal weight to all three chunks. For GerTE, in contrast, we assigned less weight to the similarity score of the top chunk pair since most essays in GerTE start by providing meta data on the source article. We considered similarity in this area to be less relevant.

We performed our experiments on both datasets using cross-validation, with $k \in \{1, 2, 4, 6, 8\}$ for GerTE and $k \in \{1, 2, 4, 6, 8, 10\}$ for FDE-Arg. If k is larger than the number of candidate examples available¹⁰, all candidates were selected and used.

4.4 Parameter-Efficient Fine-tuning

As an alternative to improving the prompt fed to the LLMs, we also experimented with parameter-efficient fine-tuning (PEFT) using QLoRA (Dettmers et al., 2023) with 4-bit quantization. Due to computational costs, this approach was only applied to the smaller model Llama-8B. We again divided our data into training and test partitions through 5-fold cross-validation and fine-tuned the model on the training partition. For either dataset, the model was fine-tuned on the zero-shot baseline prompt text, paired with the target output, which could be derived from the gold-standard zone labels in the datasets. The approach was implemented using the PEFT functionalities that are integrated into the Huggingface framework. All hyperparameters used are provided in the Appendix.

¹⁰This can happen if the training portion in a given fold does not contain sufficient essays on the same topic as the target essay.

5 Results

For both datasets, we adopted a regex-based method of extracting labels from the LLM’s output. Where the model’s output used made-up zone labels or where it did not produce the same number of labels as the number of input segments, we considered the output invalid and excluded it from evaluation.

Based on all test essays that yielded valid output, we evaluated model performance using standard multi-class classification metrics, viz. accuracy and average F1. For comparability, we follow Bai and Stede (2025) and used the *weighted* average F1 for GerTE; for FDE-Arg, we saw no justification for assigning frequency-dependent weights to the different content zones and therefore used the *macro* average F1.

Our experiment results are summarised in Table 4 for GerTE¹¹ and in Table 5 for FDE-Arg¹². Where cross-validation was used, the metrics are as averaged across all folds. The best-performance on either dataset by either model is printed in bold.

Our results reveal the following general trends: First, in line with expectations, the larger Llama model with 70 billion parameters consistently outperforms the smaller one. On GerTE, to which the larger model has not yet been applied, it achieves a weighted F1 of 0.79 under the best experimental condition (similarity-based six-shot prompting). This by far beats the hitherto highest F1 of 0.713, which is the performance of the best supervised model reported by Bai and Stede (2025), while requiring only six labelled essays. Moreover, with the same six-shot prompting scheme, the smaller Llama-8B model in fact also surpasses said supervised model, achieving an F1 of 0.73.

Second, confirming Bai et al. (2026)’s observations, zone extraction on FDE-Arg proves to be highly challenging, particularly for the smaller Llama-8B model. In fact, it produced invalid outputs for one-third to half of the test essays, and the labelling results on those that did receive valid outputs were consistently poor. Upon manual examination of the invalid outputs, we found that nearly all of them are cases of made-up labels that were simply not the ones that the model had been

¹¹Here, the results for the baselines using Llama-8B are taken from Bai and Stede (2025)’s original paper.

¹²Note that the baseline performance cannot be directly compared to the results in Bai et al. (2026) since their experiments only used 50 essays rather than our 100-sample FDE-Arg.

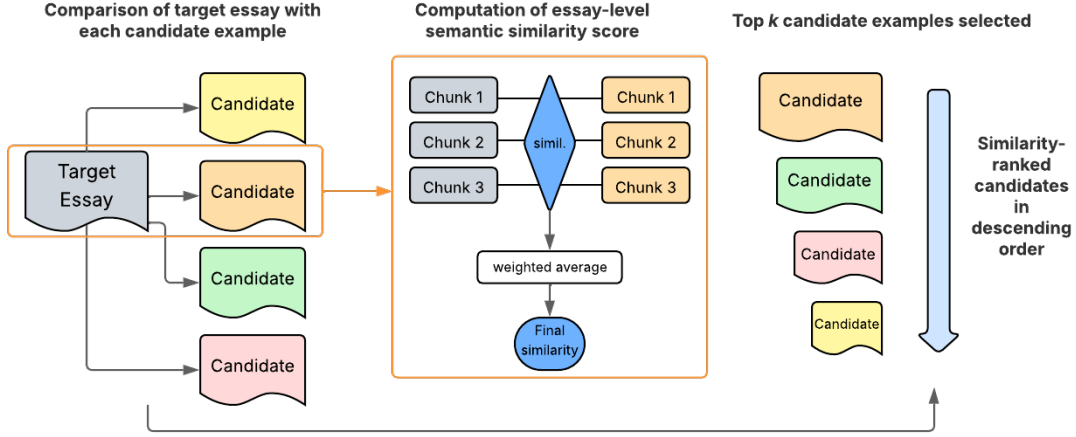


Figure 2: Ranking of candidate example essays for a given target essay based on semantic similarity.

	Llama-8B			Llama-70B		
	# invalid	Acc	F1 (w-avg)	# invalid	Acc	F1 (w-avg)
zero-shot (baseline)	12	0.654	0.641	1	0.736	0.731
zero + article (baseline)	8	0.673	0.661	0	0.755	0.745
one-shot (baseline)	10	0.661	0.649	0	0.751	0.743
zero + prompt addition (article + arguments)	8	0.686	0.672	0	0.776	0.77
one + prompt addition (article + arguments)	7	0.687	0.680	0	0.78	0.774
Similarity-based few-shot prompting						
$k = 1$	6	0.703	0.686	1	0.762	0.748
$k = 2$	7	0.707	0.691	0	0.766	0.751
$k = 4$	17	0.716	0.702	0	0.772	0.755
$k = 6$	9	0.742	0.73	0	0.8	0.79
$k = 8$	12	0.732	0.721	1	0.782	0.762
PEFT	2	0.635	0.638	-	-	-

Table 4: Content zone prediction results on GerTE in terms of accuracy (Acc) and weighted-average F1 (F1, w-avg). “# invalid” refers to the number of test essays which were excluded from evaluation due to invalid output.

instructed to use. The same behaviour of generating illegal labels is also observed with Llama-8B on GerTE, though much less frequently (see Table 4). Overall, the F1 scores achieved by Llama-8B should be interpreted with the number of invalid outputs in mind and are arguably less conclusive.

Finally, contrary to expectations, our fine-tuning experiments using QLoRA on Llama-8B did not bring about any improvements in terms of the chosen metrics. Our fine-tuned models performed near or even below the base models. We do observe, nonetheless, that on both datasets, the fine-tuning did lead the model to produce far fewer instances of invalid output. This is particularly true for FDE-Arg (see Table 5).

To illustrate common challenges in our task, Figure 3 shows the truth-normalised confusion matrices (values within rows add up to 1) for the best-performing model, which, for either dataset, is Llama-70B in a similarity-based few-shot prompt

scenario. The values are as averaged across all folds in cross-validation. For GerTE, “other” stands out as the most difficult class to recognise. As pointed out by Bai and Stede (2025), this is expected since it is both the least frequent label and a fuzzily defined fall-back class for all sentences that do not fit elsewhere. For FDE-Arg, confusion between “th1” and “th2” and, concurrently, between “pro” and “con” form the main challenges.

6 Discussion and Further Analysis

The focus of our experiments lies in exploring the three aforesaid strategies for improving LLM performance on the content zone prediction task. Based on our results, similarity-based few-shot prompting with approximately 6 to 8 labelled essays is particularly promising, showing the best prediction results on both datasets. This raises the question whether this success can be attributed to the sheer use of more example essays or to the

	Llama-8B			Llama-70B		
	# invalid	Acc	F1 (m-avg)	# invalid	Acc	F1 (m-avg)
zero-shot (baseline)	45	0.51	0.380	1	0.576	0.490
two-shot (baseline)	41	0.628	0.514	1	0.745	0.632
zero + “cth” clarification	48	0.525	0.397	3	0.618	0.529
two + “cth” clarification	48	0.593	0.448	4	0.684	0.640
zero + recommended prediction steps	44	0.421	0.329	1	0.588	0.530
two + recommended prediction steps	41	0.607	0.482	1	0.686	0.649
Similarity-based few-shot prompting						
$k = 1$	39	0.549	0.446	0	0.641	0.585
$k = 2$	46	0.567	0.476	1	0.708	0.66
$k = 4$	43	0.599	0.471	1	0.749	0.698
$k = 6$	38	0.612	0.519	2	0.728	0.685
$k = 8$	38	0.602	0.499	0	0.773	0.750
$k = 10$	36	0.597	0.504	1	0.76	0.729
PEFT	6	0.547	0.408	-	-	-

Table 5: Content zone prediction results on FDE-Arg in terms of accuracy (Acc) and macro-average F1 (F1, m-avg). “# invalid” refers to the number of test essays which were excluded from evaluation due to invalid output.

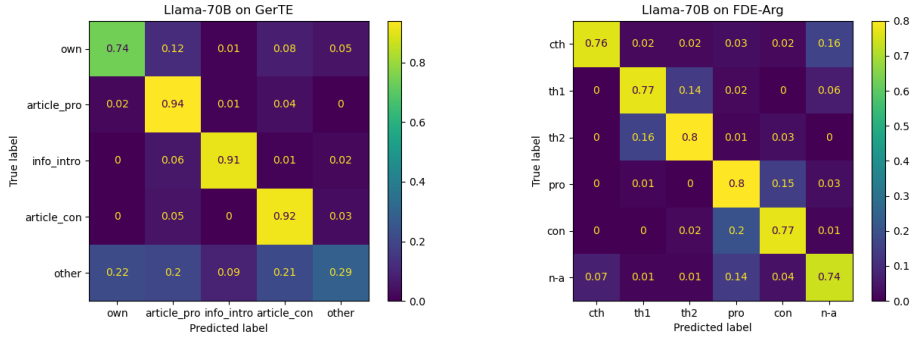


Figure 3: Truth-normalised confusion matrices for the best-performing experimental setting on either dataset.

semantic similarity between example and target essays. To further examine this, we also experimented with k -shot prompting with *random* example selection. In this case, k examples are simply chosen at random from the set of candidate essays, viz. training essays that address the same topic as the target essay.

Appendix C provides the detailed F1 scores achieved by the models using randomly selected example essays, once again as averaged across all folds in cross-validation. Figure 4 graphically compares the F1 scores¹³ obtained in the similarity-based vs. the random few-shot prompting scheme. We have left out the performance of Llama-8B on FDE-Arg, where the large amounts of invalid outputs render the results inconclusive.

Overall, in both similarity-based and random few-shot prompting, an improvement of model performance can be observed with increasing numbers of examples k until it plateaus at approximately $k = 6$ or $k = 8$. While choosing examples based

on semantic similarity with target samples does not outperform choosing them randomly across all values for k , the similarity-based scheme is superior in the majority of the cases, especially with the application of Llama-70B to FDE-Arg. Moreover, across the board it is the similarity-based prompt setting that produces the best performance for a given model-dataset combination.

While not as effective as few-shot prompting, our experiments have also shown that adding task-relevant information to the prompt text can indeed benefit LLMs’ zone prediction performance. This is especially attractive since this strategy is highly cost-efficient, requiring neither significantly more computational resources (unlike PEFT) nor the availability of labelled training data. As shown by Table 4 and Table 5, in all of our experiments, the inclusion of our chosen additional information in the prompt text produced performance improvement compared to the baseline settings. The only exception here is that of Llama-8B on FDE-Arg, which, again, is less conclusive due to the abundance of invalid output. With respect to GerTE,

¹³Once again, *weighted-average* F1 for predictions on GerTE, *macro-average* F1 for those on FDE-Arg.

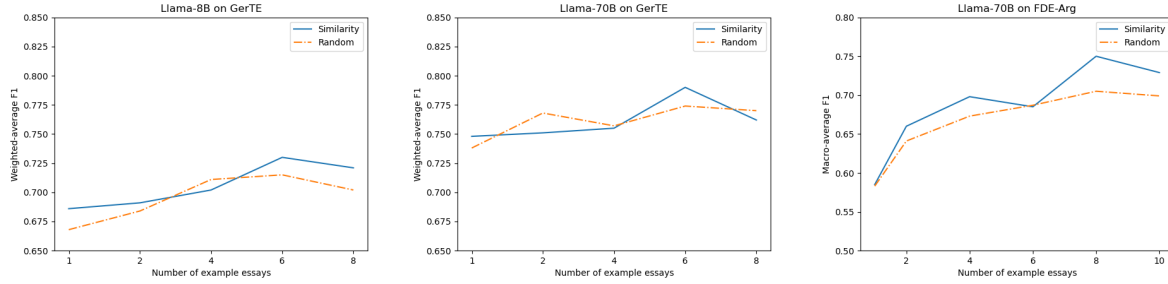


Figure 4: Graph representation of the F1 scores in the similarity-based (solid blue line) vs. the random (dashed orange line) k -shot prompting scheme.

the inclusion of our summary of the source article arguments primarily aims at improving the model’s performance on the classes “article_pro”, “article_con” and “own”. Similarly, the annotation guide-inspired additions to the prompt text for FDE-Arg have been designed to target correct prediction of the classes “cth”, “th1” and “th2”. Table 6 and Table 7 respectively show the class-specific F1 values as achieved by Llama-70B for the targeted classes in either dataset, contrasting the baselines with the settings including additional information in the prompt. It can be observed that the targeted classes have benefitted from the additions almost across the board.

Label	a_pro	a_con	own
zero (baseline)	0.788	0.701	0.766
zero + article + argument	0.869	0.777	0.772
one (baseline)	0.804	0.694	0.793
one + article + arguments	0.867	0.761	0.799

Table 6: Llama-70B on GerTE: Class-specific F1 for “article_pro” (“a_pro”), “article_con” (“a_con”) and “own” in selected experiment conditions.

Label	cth	th1	th2
zero (baseline)	0.454	0.26	0.392
zero + “cth” clarification	0.496	0.27	0.444
zero + prediction steps	0.476	0.333	0.536
two (baseline)	0.561	0.522	0.618
two + “cth” clarification	0.554	0.532	0.639
two + prediction steps	0.576	0.557	0.657

Table 7: Llama-70B on FDE-Arg: Class-specific F1 for “cth”, “th1” and “th2” in selected experiment conditions.

Finally, as remarked in the previous section, fine-tuning using PEFT and QLoRA has not proved useful in our experiments. We do note, however, that both datasets we used are small for the purpose of model training, and that we have not performed any hyperparameter tuning. We therefore consider this result to be tentative and in need of further

experimentation.

7 Conclusion

We have introduced the novel German dataset FDE-Arg, which consists of 100 stand-alone argumentative essays by secondary school students and are annotated with a fine-grained set of argumentative content zones as developed by Bai et al. (2026). Using both FDE-Arg and source-dependent essays from the GerTE dataset, we applied two differently-sized Llama models to the task of predicting the argumentative content zones. While promising results could be obtained from the larger model, the smaller model struggled with the task, especially with the more challenging FDE-Arg dataset.

We further focused on three strategies for boosting model performance and found both the simple incorporation of additional, task-related information into the prompt text and few-shot prompting with approximately 6 to 8 labelled example instances to be effective. Moreover, choosing the examples based on semantic similarity with the target instance seems to be particularly promising. We could not observe any performance improvement through our current implementation of the fine-tuning approach using PEFT. However, FDE-Arg is gradually being expanded, and argumentative zone labels are being applied to a growing portion of the original FDE dataset. Therefore, we plan to revisit PEFT-based strategies in the future using significantly more labelled data and newer LLMs. Moreover, given the success of our similarity-based few-shot prompting scheme, we also plan to explore alternative ways to capture similarity between target and example instances in order to improve example selection for in-context learning.

Limitations

We acknowledge the following limitations in relation to our work: First, both datasets we used are rather small. GerTE contains 117 essays, while FDE-Arg contains 100. However, as mentioned in the main text, FDE-Arg is growing, and the original FDE dataset consists of 1,061 essays, which we plan to fully annotate with argumentative content zones within the next months. We are therefore confident that we will soon be able to elaborate on our experiments with more labelled data. Second, our experiments with PEFT have only been a first step, and we did not perform hyperparameter tuning. We plan to look more into fine-tuning-based approaches using LLMs once more training data from FDE-Arg are available. Finally, for now, with Llama 3 we have chosen a well-known but comparatively old LLM since our focus does not lie in comparisons between different LLMs. We do plan to repeat our experiments with more up-to-date LLMs in the future.

Acknowledgments

We thank the anonymous reviewers for their highly helpful comments and suggestions on the first draft of this paper. We are also grateful to Jun. Prof. Winnie-Karen Giera and her team for building the original FDE corpus and cooperating with us. Kemal Afzal and Dietmar Benndorf have made significant contributions to the annotation scheme for FDE-Arg. Part of the work reported here is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 567969163.

References

- Yamen Ajjour and Henning Wachsmuth. 2025. [Exploring LLM Priming Strategies for Few-Shot Stance Classification](#). In *Proceedings of the 12th Argument mining Workshop*, pages 11–23, Vienna, Austria. Association for Computational Linguistics.
- Xiaoyu Bai, Kemal Afzal, Dietmar Benndorf, Lucas Deutzmann, Winnie-Karen Giera, Eric Graßnick, and Manfred Stede. 2026. From newspapers to classrooms: Adapting an annotation scheme and automatic classifiers to mixed-quality argumentative school essays. Submitted.
- Xiaoyu Bai and Manfred Stede. 2022. [Argument Similarity Assessment in German for Intelligent Tutoring: Crowdsourced Dataset and First Experiments](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2177–2187, Marseille, France. European Language Resources Association.
- Xiaoyu Bai and Manfred Stede. 2023. [A Survey of Current Machine Learning Approaches to Student Free-Text Evaluation for Intelligent Tutoring](#). *International Journal of Artificial Intelligence in Education*, 33(4):992–1030.
- Xiaoyu Bai and Manfred Stede. 2025. [Predicting Functional Content Zones in German Source-Dependent Argumentative Essays: Experiments on a Novel Dataset](#). In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Long and Short Papers*, pages 66–79, Hannover, Germany. HsH Applied Academics.
- 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.
- Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. 2025. [Argument Mining with Fine-Tuned Large Language Models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6624–6635, Abu Dhabi, UAE. Association for Computational Linguistics.
- Guizhen Chen, Liying Cheng, Anh Tuan Luu, and Li-dong Bing. 2024. [Exploring the Potential of Large Language Models in Computational Argumentation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2309–2330, Bangkok, Thailand. Association for Computational Linguistics.
- S. A. Crossley, Y. Tian, P. Baffour, A. Franklin, M. Benner, and U. Boser. 2024. [A large-scale corpus for assessing written argumentation: PERSUADE 2.0](#). *Assessing Writing*, 61:100865.
- 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.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- W. K. Giera, M. Stede, L. Deutzmann, and E. Graßnick. 2025a. Exploring the Power of Persuasion in Written Argumentation: A Mixed-Methods Pilot Study (QASA). *Journal of Applied Language Learning*, 2(2):1–10.
- Winnie-Karen Giera, Lucas Deutzmann, and Subhan Sheikh Muhammad. 2025b. Merging oral and written argumentation: Supporting student writing through debate and srtd in inclusive classrooms. *Education Sciences*, 15(11):1471.
- Deniz Gorur, Antonio Rago, and Francesca Toni. 2025. [Can Large Language Models perform Relation-based Argument Mining?](#) In *Proceedings of the 31st International Conference on Computational Linguistics*,

- pages 8518–8534, Abu Dhabi, UAE. Association for Computational Linguistics.
- Masayuki Kawarada, Tsutomu Hirao, Wataru Uchida, and Masaaki Nagata. 2024. [Argument Mining as a Text-to-Text Generation Task](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2002–2014, St. Julian’s, Malta. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. [What makes good in-context examples for GPT-3?](#) In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks](#). *arXiv:1908.10084 [cs]*. ArXiv: 1908.10084.
- Nils-Jonathan Schaller, Yuning Ding, Andrea Horbach, Jennifer Meyer, and Thorben Jansen. 2024a. [Fairness in Automated Essay Scoring: A Comparative Analysis of Algorithms on German Learner Essays from Secondary Education](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 210–221, Mexico City, Mexico. Association for Computational Linguistics.
- Nils-Jonathan Schaller, Andrea Horbach, Lars Höft, Yuning Ding, Jan Luca Bahr, Jennifer Meyer, and Thorben Jansen. 2024b. [DARIUS: A Comprehensive Learner Corpus for Argument Mining in German-Language Essays](#).
- Christian Stab and Iryna Gurevych. 2017. [Parsing Argumentation Structures in Persuasive Essays](#). *Computational Linguistics*, 43(3):619–659.
- Maja Stahl, Nadine Michel, Sebastian Kilsbach, Julian Schmidtke, Sara Rezat, and Henning Wachsmuth. 2024. [A School Student Essay Corpus for Analyzing Interactions of Argumentative Structure and Quality](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2661–2674, Mexico City, Mexico. Association for Computational Linguistics.
- Simone Teufel et al. 1999. *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, University of Edinburgh Edinburgh, Scotland.
- Thiemo Wambsganss, Christina Niklaus, Matthias Cetto, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020a. [AL: An Adaptive Learning Support System for Argumentation Skills](#). In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14. Association for Computing Machinery, New York, NY, USA.
- Thiemo Wambsganss, Christina Niklaus, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020b. [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.
- 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.

A LLM Prompts

The following shows a baseline prompt used on FDE-Arg, featuring an authentic target essay from the dataset.

System prompt:

Du bist ein Deutschlehrer der 9. Klasse und analysierst argumentative Aufsätze deiner Schüler. Die Schüler schreiben den Aufsatz mit Bezug auf eine vorgegebene Streitfrage. Du überprüfst, wie der Aufsatz argumentativ aufgebaut ist.

User prompt:

Im Folgenden liegt ein Aufsatz vor. Der Aufsatz wurde in kleineren Einheiten wie Sätzen oder Phrasen unterteilt, die im folgenden zur Vereinfachung als 'Sätze' bezeichnet werden. Die Sätze sind nummeriert. Das Format dabei ist 'Satznummer: Satz'.
Entscheide zunächst, welche argumentative Gesamtkonstellation vorliegt:
'Entschieden' bedeutet: Der Aufsatz argumentiert für eine bestimmte Position des Autors.
'Unentschieden' bedeutet: Der Aufsatz bringt gleichberechtigt Argumente für beide Seiten der Streitfrage hervor. Der Autor entscheidet sich nicht für eine Position.
Analysiere die Funktionen der einzelnen Sätze. Dabei gilt eine grundlegende Unterscheidung zwischen der Funktion der These und der des Arguments: Eine These ist eine argumentative Position, die jemand einnehmen kann. Ein Argument ist ein Grund, der eine bestimmte These stützt oder angreift.
Jeder Satz im Aufsatz soll einer der folgenden 6 Funktionen zugeordnet werden:
'Zentrale_These' bedeutet: Der Satz beschreibt in einem entschiedenen Aufsatz die Kernposition

des Autors, oder er drückt in einem unentschiedenen Aufsatz explizit aus, dass der Autor sich nicht für eine Position entscheiden kann.

'These_1' bedeutet: Der Satz beschreibt in einem entschiedenen Aufsatz die Position, die mit der zentralen These übereinstimmt, oder er beschreibt in einem unentschiedenen Aufsatz die im Text erstgenannte Position.

'These_2' bedeutet: Der Satz beschreibt in einem entschiedenen Aufsatz die Position gegen die der zentralen These, oder er beschreibt in einem unentschiedenen Aufsatz die im Text später genannte Position.

'Pro_Argument' bedeutet: Der Satz beschreibt ein Argument, das These 1, also die Position des Autors in einem entschiedenen Aufsatz, stützt und bestärkt.

'Con_Argument' bedeutet: Der Satz beschreibt ein Argument, das These 2, also die Position gegen die des Autors in einem entschiedenen Aufsatz, stützt und bestärkt.

'Nicht_Argumentativ' bedeutet: Der Satz hat keine argumentative Funktion. Stattdessen gibt er zum Beispiel Hintergrundinformationen zum Thema, Anekdoten des Autors oder dient der Gliederung und der abschließenden Zusammenfassung des Aufsatzes.

Werte den Aufsatz aus und ordne jeden Satz einer der genannten 6 Funktionen zu. Das Format des Outputs soll ausschließlich wie folgt sein:

'Gesamtkonstellation

Satznummer: Funktion

Satznummer: Funktion'

, zum Beispiel

'Entschieden

1: Zentrale_These

2: Nicht_Argumentativ

3: Pro_Argument

4: These_1'.

Im Output sollen es neben der Gesamtkonstellation genauso viele Satznummer-Funktion-Paare geben wie es Sätze im Aufsatz gibt.

Schüleraufsatz

1: Was ist nun besser, Fast Food oder selbstgekochtes, gesundes Essen?

2: Immer öfter findet eine Diskussion zwischen den beiden Seiten statt, in der über die bessere Art von Essen debattiert, doch was ist nun besser?

3: Heutzutage konsumieren Kinder oder Teenager immer mehr Fast Food,

4: es ist schnell, günstig und lecker.

5: Dies stellt eine große Erleichterung für sie dar, wenn sie zum Beispiel nach einem stressigen Schultag schnell in ein Fast Food Restaurant gehen können und innerhalb kürzester Zeit ihr Essen vor sich stehen haben.

6: Jedoch kommen auch Argumente ans Licht die definitiv gegen eine solche Ernährung sprechen und eine gesündere Ernährung befürworten.

7: Aus diesem Anlass wird in dieser Erörterung über die positiven Seiten beider Essensarten, aber auch über ihre Probleme gesprochen.

8: Selbst gekochtes Essen ist viel gesünder

9: und besser als Fast Food,

10: da es eine bessere Essensqualität hat, das steht nicht zur Frage.

11: Man nimmt weniger Zusatzstoffe und Kalorien zu sich,

12: da man frische Sachen wie Gemüse verwendet.

13: Selbstkochen kann Spaß machen

14: und man findet vielleicht ein neues Hobby am Kochen,

15: doch auch Fast Food hat seine Vorteile.

16: Im Gegensatz zum selbstgekochten gesunden Essen, ist dieses günstiger und schneller,

17: fast immer schmeckt es einem

18: und nicht jeder hat noch Energie um nach einem anstrengendem Tag m Herd zustehen und zu kochen.

19: Fast Food findet man ja auch heutzutage überall, wenn man zum Beispiel in die Stadt geht um zu shoppen oder einkaufen zu gehen.

20: Alle Seiten haben ihre Vorteile aber auch Nachteile vorallem im Thema Gesundheit, wo frisch zubereitetes Essen auf jedenfall besser für den Körper ist doch man kann beide Seiten verstehen.

21: Zusammenfassend kan man sagen, dass Fast Food natürlich deutlich weniger Aufwand macht, zudem günstiger ist, aber auf eine gesunde Ernährung kann man nicht verzichten.

22: Meiner Meinung nach sollte man sich in der Mitte treffen.

23: Selbstgekochte, gesündere Sachen sind zwar besser und tragen zu einer guten Ernährung bei, aber man kann sich ab und zu einen Burger gönnen.

24: Doch übertreiben sollte man es nicht.

25: Zum Schluss lässt sich sagen, dass jeder das essen soll, was er/sie am besten für sich selber hält, aber eine gesunde Ernährung schaded nie.

The following shows a system prompt on GerTE with the added summary of source article arguments (see Section 4.2.1). The user prompt is the same as in the baseline prompt scenario.

Du bist ein Deutschlehrer der 9. Klasse und analysierst Aufsätze deiner Schüler. Du überprüfst dabei, ob die Aufsätze alle zu erwartenden Bestandteile enthält.

Hier ist der Lesetext, auf den der Aufsatz Bezug nimmt.

[FULL SOURCE ARTICLE TEXT]

Hier sind die Argumente, die im Lesetext vorkommen: Die Pro-Argumente sind folgende: Unterricht ist abwechslungsreicher.

Unterricht bleibt länger im Gedächtnis.

Nutzung von Twitter senkt die Hemmung vor aktiver Beteiligung am Unterricht.

Auch schüchterne Schüler:innen werden ermutigt, ihre Meinungen zu äußern.

Pädagogen raten zur Auseinandersetzung mit dem medialen Wandel.

Es gibt Erfolgsbeispiele für die Nutzung von Twitter im Unterricht, z.B. in den USA.

Angst vor Ablenkung ist nicht begründet, da auch Bücher ablenken können.

Laut Psychologen fördert das Internet die Gesprächskultur.

Die Con-Argumente sind folgende:

Schüler können leicht abgelenkt werden.

Viele Lehrer sind neuen Medienformen gegenüber skeptisch eingestellt.

The following shows the addition to the prompt text for FDE-Arg concerning the distinction between “cth” and “n-a” (see Section 4.2.2).

Manche Aufsätze enden mit einer knappen Zusammenfassung der Position des Autors, zum Teil etwas umformuliert. Sofern die zentrale These bereits woanders markiert wurde, wird dieser Abschlusssatz als Zusammenfassung des Aufsatzes gewertet und somit als “Nicht_Argumentativ” markiert. Gibt es hingegen an keiner anderen Stelle die zentrale These, so wird dieser Abschlusssatz als “Zentrale_These” markiert.

The following shows the addition to the FDE-Arg prompt text concerning the recommended prediction steps (see Section 4.2.2).

Bei der Zuordnung kannst du wie folgt vorgehen:

1. Finde die Kernaussage des Textes (“Zentrale_These”): Diese steht für sich selbst und kann prinzipiell an jeder Stelle des Textes stehen.
2. Finde weitere Thesen (“These_1” / “These_2”): Wenn der Text beide Positionen beleuchtet, unabhängig davon, ob der Autor sich letztendlich für eine Seite entscheidet, werden die beiden Positionen markiert.
3. Finde Argumente (“Pro_Argument” / “Con_Argument”): Argumente begründen die Thesen. Dabei können sie die Thesen direkt stützen oder ein anderes Argument und damit indirekt die Thesen stützen.
4. Markiere verbleibende Sätze (“Nicht_Argumentativ”): Alle Sätze, die zuvor nicht markiert wurden, haben keine argumentative Funktion.

B PEFT Hyperparameters

Hyperparameters for the PEFT-experiments on either dataset are given in Table 8. All parameters not mentioned in the table used the default values by HuggingFace.

C Few-Shot Prompting with Randomly Selected Examples

Table 9 and Table 10 give the experimental results on few-shot prompting using *randomly* selected example essays. All values are as averaged across all folds in cross-validation.

	GerTE	FDE-Arg
learning rate	$5e^{-5}$	$2e^{-4}$
per_device_train_batch_size	4	4
gradient_accumulation_steps	2	4
maximum training steps	320	300
LoRA-specific hyperparameters		
r	8	16
lora_alpha	16	16
lora_dropout	0.05	0
bias	none	none
task_type	CAUSAL_LM	CAUSAL_LM

Table 8: Hyperparameters used in the PEFT-experiments on either dataset.

	Llama-8B			Llama-70B		
	# invalid	Acc	F1 (w-avg)	# invalid	Acc	F1 (w-avg)
$k = 1$	13	0.688	0.668	1	0.755	0.738
$k = 2$	15	0.699	0.684	0	0.784	0.768
$k = 4$	7	0.726	0.711	0	0.773	0.757
$k = 6$	8	0.725	0.715	1	0.789	0.774
$k = 8$	12	0.72	0.702	0	0.787	0.77

Table 9: Prediction results on GerTE using the few-shot prompting scheme with *randomly* selected examples.

	Llama-8B			Llama-70B		
	# invalid	Acc	F1 (m-avg)	# invalid	Acc	F1 (m-avg)
$k = 1$	50	0.533	0.413	0	0.638	0.583
$k = 2$	41	0.579	0.462	0	0.697	0.641
$k = 4$	41	0.637	0.531	0	0.715	0.673
$k = 6$	37	0.565	0.466	1	0.728	0.687
$k = 8$	35	0.621	0.505	1	0.735	0.705
$k = 10$	41	0.612	0.506	1	0.741	0.699

Table 10: Prediction results on FDE-Arg using the few-shot prompting scheme with *randomly* selected examples.

Multi-step Large Language Model for Fine-Grained Feedback in Stepwise Linear Equation Solutions

Imran Chamieh¹, Torsten Zesch², Klaus Giebertmann¹

¹Hochschule Ruhr West, Germany

²CATALPA, FernUniversität in Hagen, Germany

Abstract

This paper addresses the problem of fine-grained error classification in stepwise algebraic problem solving, with the objective of enabling precise and timely feedback in large-scale educational environments. Using authentic student data, we compare a carefully engineered rule-based baseline with large language models (LLMs) in zero-shot and few-shot configurations, as well as multi-step LLM-based approaches. We further consider hybrid architectures that combine symbolic computation with LLM-based reasoning. Our empirical results indicate that, although the baseline model delivers strong and reliable performance for narrowly defined error categories, the proposed multi-step LLM framework surpasses the rule-based model by achieving superior precision, F1 scores, and overall accuracy. These findings underscore the limitations of unstructured end-to-end prompting for error classification.

1 Introduction

Digital mathematics learning platforms routinely mark intermediate solution steps as correct or incorrect yet often fail to explain *why* a step is wrong. Whereas short feedback can lead the student to the correct solution. Consider the following example from Figure 1

$$\begin{aligned} S_2 : 6q - 4 &= 3 \\ S_3 : 6q &= 3 - 4 \end{aligned}$$

For a human instructor, it is readily apparent that the student made a sign error in the step S_3 by failing to change the sign of -4 when transposing it to the right-hand side of the equation. In conventional classroom settings, instructors typically identify such errors through detailed examination of students' solution processes and subsequently provide targeted, formative feedback. At a large scale, such

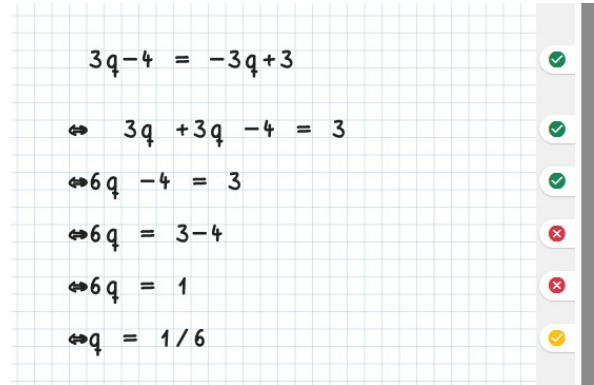


Figure 1: Example of binary error detection in an algebraic task. The system identifies incorrect steps but fails to provide explanatory feedback to the student.

as with intelligent tutoring systems (ITS), however, this process becomes increasingly difficult. ITS rely heavily on computer algebra systems (CAS) to verify whether a student's transformation is mathematically valid. While CAS reliably detects incorrect steps, it typically does not explain why a step is incorrect, so students receive binary correctness signals without diagnostic specificity, which lacks the useful feedback that allows students to self-correct their mistakes, as illustrated in 1. As a result, students may struggle to interpret their mistakes. In such cases, the educational value lies not in telling the student that the answer is wrong but in understanding *why*.

This paper examines the problem of fine-grained error classification in stepwise algebraic solutions, in contrast to the more conventional focus on assessing the correctness of complete solutions. We delineate a narrowly defined yet pedagogically meaningful error category that supports a direct mapping to targeted instructional feedback. Specifically, we address the detection of explicit algebraic sign errors ($+$ $\leftrightarrow$ $-$) in linear equations, using step-level interaction data collected from an intelligent tutoring system.

We formalize this construct as follows: given an initial algebraic equation S_i and the subsequent step S_{i+1} produced by a student, S_{i+1} is labeled True if and only if a sign inversion ($+ \leftrightarrow -$) is sufficient to render the transformation mathematically valid. In cases where multiple errors occur, the label is True if a sign inversion restores the mathematical validity of the affected term.

The central research question motivating this work is the following:

To what extent can different modeling paradigms reliably detect explicit algebraic sign errors at the step level in a manner that can be generalized to other error categories?

To address this question, we design an experimental framework that operationalizes the identification of explicit sign inversions in linear equations in accordance to the above definition. Within this framework, we compare three modeling approaches:

1. a **rule-based model** that explicitly enumerates valid algebraic transformations, matches them to student-generated steps, identifies the closest correct transformation, and then determines whether a sign error is present;
2. a **single-prompt LLM model** that frames sign-error detection as a zero-shot and few-shot classification task over (S_i, S_{i+1}) step pairs;
3. a **multi-step LLM pipeline** that decomposes the reasoning process into distinct stages (simplification, intent inference, and error classification), with each stage handled by a dedicated LLM or CAS component.

The final approach is structurally analogous to chain-of-thought reasoning in that it decomposes a complex decision into a sequence of specialized subtasks, thereby reducing the cognitive and computational burden on any single model invocation. Consequently, we refer to this configuration as a multi-step LLM pipeline composed of multiple specialized components, rather than a single monolithic prompt.

Our contributions are threefold:

1. We introduce a step-level dataset of annotated linear-equation transformations that specifically targets explicit sign errors¹.

2. We conduct a controlled comparison between symbolic (rule-based) and LLM-based methods for fine-grained algebraic error detection.
3. We propose a compositional, multi-step LLM framework designed to mitigate over-generalization and enhance diagnostic precision.

2 Related Work

Beyond the binary verification provided by standard CAS, the field has long pursued the automation of fine-grained diagnostic feedback through multiple technical paradigms. Early approaches focused on model tracing and symbolic reasoning, which depend on explicitly defined domain models to map student deviations against an “expert” path (Anderson et al., 1995). In contrast to generic computer algebra verification, these systems use expert-defined rules to infer which specific algebraic transformation a student likely attempted. Although such rule-based classifiers can attain high precision under controlled conditions (McTavish and Larusson, 2014), they are commonly constrained by a pronounced “authoring bottleneck” (Bent et al., 2019). Systematically encoding the full space of student misconceptions as rigid symbolic rules is highly labor-intensive and does not scale with increasing problem complexity or heterogeneity in student behavior.

With the advent of data-driven methodologies, researchers address the limitations of manual rule engineering by adopting supervised learning-based classifiers trained on labeled student solutions. Recent work ranges from statistical diagnostic models for algebra misconceptions, like McNichols et al. (2023) works, to neural architectures that categorize fine-grained algebraic error types from stepwise solutions or written responses (Gomes and Jaques, 2020; Russell et al., 2009; Kirve et al., 2025). While these models reduce the inflexibility associated with rule-based systems, they function as “black-box” predictors and require substantial quantities of labeled training data. This requirement poses a significant limitation in contexts involving rare or highly specialized mathematical misconceptions, for which annotated corpora are typically scarce.

Recent advancements in LLMs have introduced a new paradigm, offering a balance between the structured reasoning capabilities traditionally associated with symbolic systems and the flexibility

¹<https://github.com/Imrancham/Step-Level-Error-Detection-in-Linear-Equations>

ID	S_i	S_{i+1}	Sign Error
1	$6q - 4 = 3$	$6q = 3 - 4$	True
2	$6q - 4 = 3$	$6q = -1$	False
3	$5x + 5 = x - 1$	$4x = 4$	False
4	$y + x = -x + 2$	$y = 2$	False
5	$q(h - y) = 0$	$qh + qh = 0$	True
6	$2a + 4 = 0$	$2x = -4$	False
7	$-3(x - 1) = 0$	$-3x = 3$	True
8	$x + 2 = 3$	$x = 5$	False

Table 1: Sample of equation pairs (S_i, S_{i+1}) that have different error types.

of data-driven models, while reducing the need for the huge datasets typically required by purely data-driven systems. McNichols et al. (2024) investigated the zero-shot diagnostic capabilities of LLMs such as GPT-4 in educational contexts. A recurring observation in this literature is the presence of a “precision gap” namely, the tendency of LLMs to overgeneralize when fine-grained diagnostic specificity is needed (Chamieh et al., 2024). Our work addresses this by moving away from unstructured, end-to-end prompting toward a multi-step framework that reintroduces symbolic-like rigor into the multi-step LLM process.

3 Dataset

The dataset described in this study was collected from a digital mathematics learning platform used in undergraduate engineering courses at Hochschule Ruhr West (Ruhr West University of Applied Sciences). Approximately 144 first- and second-semester students used the platform as part of mandatory mathematics courses. The platform generates individualized exercises using parameterized mathematical templates and records student solutions at the step level. Whenever a submitted step does not follow validly from the preceding one, the system stores a record comprising the previous equation (S_i) and the student’s produced next step (S_{i+1}). Although the platform detects the occurrence of an error, it does not provide a classification of the underlying error type and, hence, no targeted feedback.

3.1 Data Filtering and Annotation

Since the winter semester of 2022/23, 49,661 incorrect solution steps have been collected. These range from basic arithmetic slips to complex conceptual errors in trigonometry or calculus. For the purposes of the present study, we restrict our focus to linear equations. We filtered the raw logs to exclude entries containing non-linear terms (e.g.,

higher-degree polynomials, square roots, or variables in denominators), resulting in a subset of 4,024 equation pairs.

Each pair (S_i, S_{i+1}) was manually annotated by a domain expert actively involved in teaching the corresponding courses, with a binary label indicating the presence (True) or absence (False) of an **explicit sign error**.

Note that, according to our definition of sign error in the introduction, we exclude arithmetic mistakes in which a student carries out a numerical computation incorrectly while otherwise maintaining valid algebraic reasoning. For instance, in the second example in Table 1, the step $S_{i+1} : 6q = -1$ is not classified as a sign error, because no single modification of a sign would convert the transformation into a correct one, even though the incorrect result stems from an erroneous evaluation of $3 - 4$, in which the term -4 appears with an incorrect sign. In contrast, in the first example in Table 1, the step yielding $S_{i+1} : 6q = 3 - 4$ is categorized as an explicit sign error, since changing the minus in -4 to a plus would render the equation correct. Overall, there are 780 of the annotated instances that exhibit such explicit sign errors, which makes approximately 19%.

3.2 Dataset statistics

Figure 2 provides a visualized distribution of the dataset. We observe that student-produced equations S_{i+1} are, on average, shorter than the source equation S_i , reflecting term aggregation during manipulation. A slight increase in the number of distinct variables is observed in S_{i+1} , can be seen in the Figure 2 on the right, primarily due to handwriting-recognition artifacts (e.g., confusion between α and a , ω and w , or 0 and o).

4 Experimental Setup

This section describes the models and configurations used to identify explicit sign errors in step-wise linear equation transformations. Figure 3 provides an overview of the processing pipeline shared across approaches.

Models Evaluated To ensure our findings generalize across different architectures, we evaluate both commercial and open-weight large language models across all single-step and multi-step configurations. We consider the following models: gemini-2.5-flash, gpt-5-mini, mistral-large-latest, and Llama-3-70b.

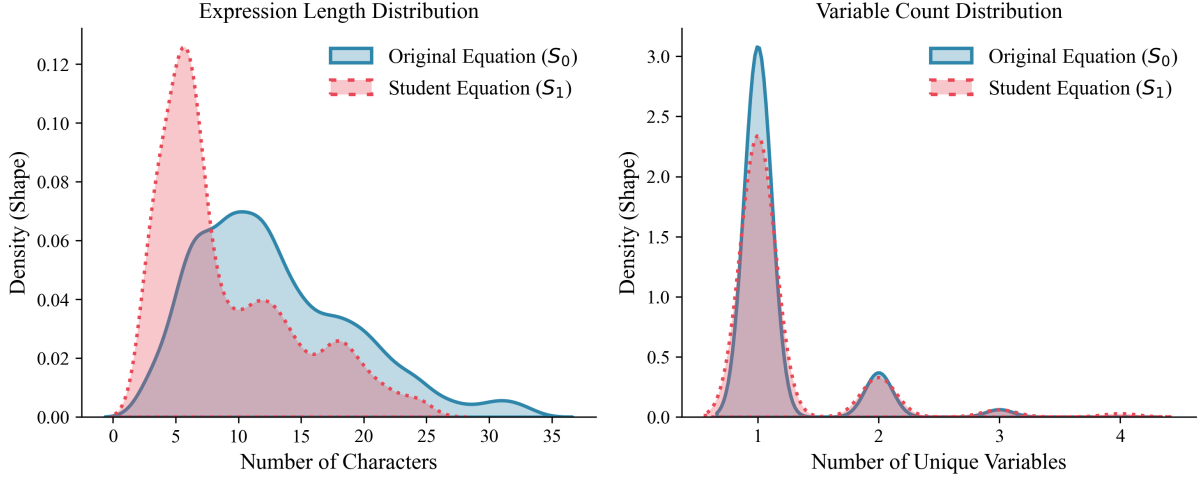


Figure 2: Distributional statistics for the dataset

Models are accessed via their respective APIs using default decoding parameters (temperature and sampling). Llama-3-70b is additionally evaluated locally on a machine equipped with 8 NVIDIA Quadro RTX 8000 GPUs, using the same generation settings to mirror the API configuration. To mitigate run-to-run variability inherent in generative models and to ensure stable performance estimates, all LLM-based experiments are repeated five times, and we report the mean and standard deviation performance across these runs.

4.1 Rule-Based Baseline

We implement a rule-based model as a high-precision and fully interpretable baseline for sign error detection. Rule-based systems allow explicit control over error definitions and decision criteria, which is particularly important in educational feedback settings where false positives can negatively affect student trust. This model also serves as a reference point for evaluating the behavior of LLM-based approaches. The pipeline consists of three steps.

Transformation Enumeration Given an initial equation S_i , we construct a transformation tree that enumerates all possible subsequent algebraic operations, including term transposition, coefficient normalization via division, distribution, and progressive variable isolation. Each node in this tree corresponds to an equation that is algebraically equivalent to S_i ; see Figure 4. This procedure establishes the foundation for inferring student intent by enabling the identification, in a subsequent stage, of the equation that is most similar to the

student’s produced step.

Similarity Matching. To approximate the student’s intended transformation, we compute a similarity score between the student’s equation S_{i+1} and each equation in the transformations tree. Prior to comparison, equations are represented in two normalized forms: (i) a structural form derived from SymPy’s internal expression representation and (ii) a canonical string form obtained by rewriting equations into a normalized representation (e.g., $a = b \rightarrow a - b = 0$). For each representation, similarity is computed using Python’s SequenceMatcher, based on the Ratcliff–Obershelp algorithm [Ratcliff et al. \(1988\)](#), which estimates the degree of overlap between two string representations by identifying matching subsequences. The final similarity score is obtained as a weighted combination of the similarity scores computed over the structural and canonical representations. These weights are fixed across all experiments. The candidate transformation with the highest combined similarity score is selected as the reference equation.

Sign Error Detection Finally, we compare S_{i+1} against the selected reference equation term by term. Matching terms are removed, and remaining terms are inspected for sign inversion. If a term in S_{i+1} differs from its counterpart in the reference equation only by the sign attached to that term, and flipping this sign would make the step valid, the pair (S_i, S_{i+1}) is classified as containing an explicit sign error.

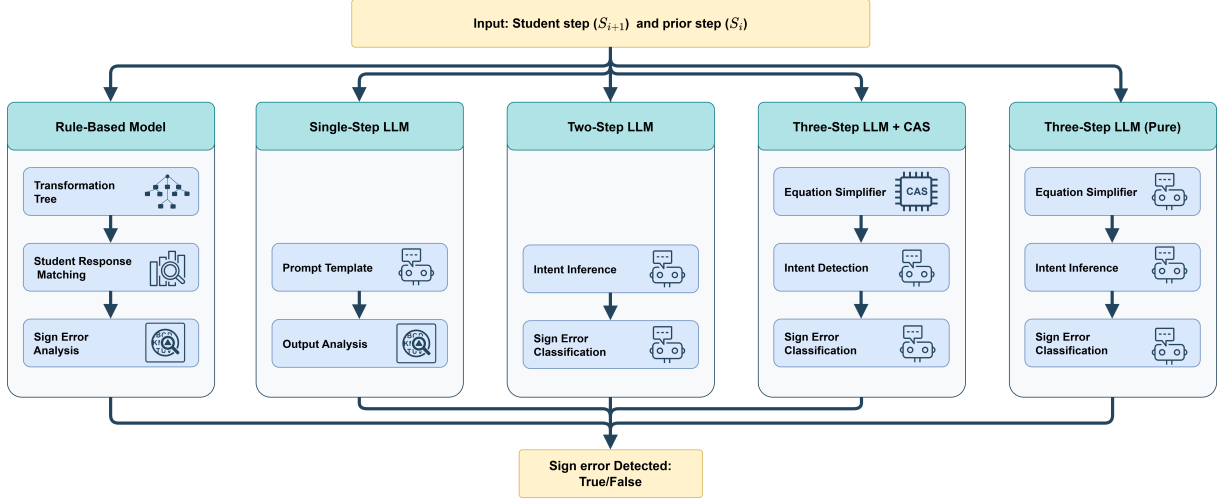


Figure 3: Comparative architecture of three diagnostic approaches—rule-based, multi-agent LLM, and prompt-based LLM—designed to identify sign errors by evaluating student input S_1 against the preceding step.

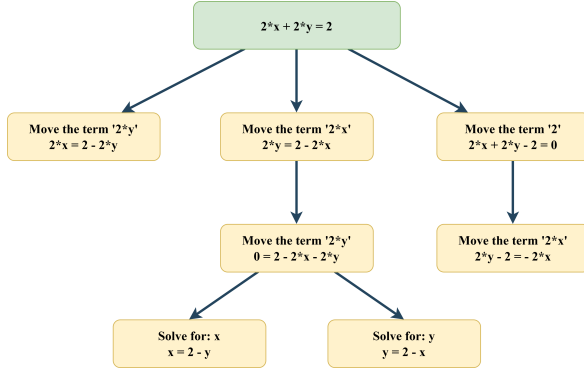


Figure 4: Tree-structured graph depicting the sequence of derived equation transformations, with the corresponding operations explicitly annotated at each transformation step.

4.2 Single-Step LLM

In the single-step paradigm, we evaluate the models as direct sign error classifiers under two distinct settings: zero-shot and few-shot. Each model receives the equation pair (S_i, S_{i+1}) along with a prompt defining explicit versus implicit sign errors in accordance with our annotation scheme. The models are instructed to output a binary decision (True or False) indicating the presence of an explicit sign error.

Prompt Engineering We experiment with multiple prompt variants that describe the task and define the notion of the explicit sign error, including different numbers of labeled examples. These prompt variants are evaluated on a subset of 200 examples of the data for each model, and we select a single best-performing prompt that is then used for all

reported results.

Output Normalization LLMs occasionally produce explanatory text, despite being instructed to output only True or False. We therefore post-process outputs using simple regular expressions to extract the final binary decision.

4.3 Multi-Step LLM

To address the precision limitations and overgeneralization observed in the single-step paradigm, we hypothesize that decomposing sign-error detection into a sequence of focused subtasks improves diagnostic reliability. Rather than relying on a single model invocation to simultaneously perform mathematical simplification, intent inference, and error classification, we distribute these operations across distinct components with structured input–output contracts. We evaluate three variations of this multi-step architecture:

1. Three-Step Pure LLM Pipeline This configuration relies entirely on LLMs for all operations:

- **Step 1 (Canonical Simplification):** An LLM agent simplifies both the original equation S_i and the student equation S_{i+1} into canonical forms by combining similar terms. Where applicable, it also computes the correct final solution for S_i .
- **Step 2 (Intent Inference and Correct Transformation):** A second LLM receives the raw and simplified forms of both equations. It infers the student’s intended algebraic operation

(e.g., transposing a term) and applies it correctly to S_i to generate a “gold” intermediate equation. It also flags cases where the student’s canonical form deviates substantially from expectations, signaling non-sign-related errors (e.g., summation errors).

- **Step 3 (Sign-Error Classification):** A final LLM agent compares the student’s raw equation with the “gold” intermediate equation from Step 2, determining if the discrepancy is strictly a single-term sign flip.

2. Hybrid CAS-LLM Pipeline To evaluate the benefit of integrating deterministic mathematical tools, this variant replaces the generative simplification step with a CAS tool. We utilize SymPy to programmatically simplify S_i and S_{i+1} and compute the correct final solution. The CAS-generated symbolic representations are then fed directly into the second step for Intent Inference and Classification in step 3, functioning exactly as described in the pure LLM pipeline.

3. Two-Step LLM Pipeline To determine whether canonical simplification is strictly necessary for accurate classification, this ablation omits Step 1 entirely. The raw equations (S_i, S_{i+1}) are passed directly to an LLM for Intent Inference to generate the “gold” intermediate step, followed immediately by the Sign-Error Classification step.

4.4 Evaluation Metrics

All models are evaluated using standard binary classification metrics, including precision (P), recall (R), F1-score (F1), and accuracy (Acc). Following common practice in educational error diagnosis, we place particular emphasis on precision, since incorrectly flagging correct solution steps or providing wrong feedback is undesirable in instructional settings. For all non-deterministic models (LLMs), reported metrics correspond to the mean over five independent runs.

5 Results and Discussion

Table 2 summarizes the performance of the rule-based baseline, the single-step LLM configurations (zero-shot and few-shot), and the proposed multi-step LLM framework for detecting explicit sign errors.

Overall Performance. The rule-based system constitutes a strong and well-balanced baseline

Model	P	R	F1	Acc
Rule-Based				
	.84	.82	.83	.93
Zero-Shot LLMs				
LLaMA	.21 $\pm$.000	.77 $\pm$.000	.33 $\pm$.000	.44 $\pm$.000
Mistral	.47 $\pm$.000	.84 $\pm$.002	.60 $\pm$.000	.80 $\pm$.000
GPT	.37 $\pm$.030	.79 $\pm$.150	.51 $\pm$.060	.80 $\pm$.000
Gemini	.52 $\pm$.001	.81 $\pm$.003	.62 $\pm$.004	.83 $\pm$.000
Few-Shot LLMs				
LLaMA	.21 $\pm$.005	.78 $\pm$.024	.34 $\pm$.008	.45 $\pm$.007
Mistral	.53 $\pm$.014	.85 $\pm$.027	.65 $\pm$.017	.83 $\pm$.009
GPT	.55 $\pm$.055	.97 $\pm$.019	.70 $\pm$.024	.84 $\pm$.006
Gemini	.52 $\pm$.017	.94 $\pm$.008	.67 $\pm$.011	.82 $\pm$.007
Two-Step LLMs				
LLaMA	.19 $\pm$.000	.95 $\pm$.006	.32 $\pm$.001	.28 $\pm$.002
Mistral	.32 $\pm$.008	.73 $\pm$.026	.45 $\pm$.013	.68 $\pm$.006
GPT	.80 $\pm$.015	.55 $\pm$.039	.65 $\pm$.021	.56 $\pm$.017
Gemini	.95 $\pm$.018	.40 $\pm$.030	.56 $\pm$.030	.89 $\pm$.005
Three-Step LLMs				
LLaMA	.29 $\pm$.159	.96 $\pm$.004	.42 $\pm$.158	.37 $\pm$.143
Mistral	.30 $\pm$.006	.82 $\pm$.019	.44 $\pm$.007	.61 $\pm$.010
GPT	.85 $\pm$.009	.81 $\pm$.026	.83 $\pm$.014	.94 $\pm$.002
Gemini	.93 $\pm$.019	.78 $\pm$.023	.85 $\pm$.026	.95 $\pm$.005
Three-Step CAS-LLMs				
LLaMA	.24 $\pm$.003	.85 $\pm$.017	.37 $\pm$.006	.49 $\pm$.005
Mistral	.24 $\pm$.002	.84 $\pm$.005	.38 $\pm$.003	.51 $\pm$.003
GPT	.87 $\pm$.009	.81 $\pm$.025	.83 $\pm$.008	.94 $\pm$.002
Gemini	.90 $\pm$.013	.82 $\pm$.018	.86 $\pm$.020	.95 $\pm$.003

Table 2: Performance comparison of rule-based, prompt-based, and multi-agent models for sign error detection.

($P = 0.84$, $R = 0.82$, $F1 = 0.83$, $Acc = 0.93$), demonstrating that symbolic approaches remain highly effective when the target error category is narrowly specified and closely aligned with a fixed set of transformation rules.

Among all evaluated approaches, the highest-performing models are the **three-step LLM configurations**, in particular GPT and Gemini, which achieve performance comparable to, and in some cases marginally surpassing, the rule-based system. Gemini attains the highest overall scores ($P = 0.93$, $F1 = 0.85$, $Acc = 0.95$), while GPT exhibits similarly strong performance ($P = 0.85$, $F1 = 0.83$, $Acc = 0.94$). These findings suggest that decomposing the task into structured intermediate steps substantially enhances LLM performance relative to single-pass inference.

Zero-shot, Few-shot, and Multi-Step LLMs:

Across all models, zero-shot and few-shot prompting display a consistent precision-recall trade-off, characterized by high recall but comparatively low precision. This pattern indicates that LLMs tend to over-predict sign errors when guided solely by

prompt-based conditioning, pointing to a misalignment between natural language-based reasoning and the formal definition of explicit sign errors.

The introduction of structure via multi-step decomposition fundamentally alters this behavior. Although two-stage pipelines remain unstable, three-step architectures systematically enhance the balance between precision and recall. This pattern indicates that an explicit factorization of the overall task into sub-components—such as simplification, intent inference, and classification—is crucial for aligning model predictions with well-defined formal error categories.

Importantly, augmenting the pipeline with a CAS component does not yield consistent performance gains, suggesting that the primary bottlenecks are not computational in nature but instead arise from difficulties in interpreting and propagating intermediate representations. Collectively, these results indicate that *structured reasoning, rather than the provision of additional training examples or external computational resources, constitutes the principal driver of the observed performance improvements.*

Comparison with Rule-Based Approach:

While the rule-based system remains a strong baseline, the proposed multi-step LLM framework exhibits several distinct advantages. First, it attains comparable or marginally superior peak performance without dependence on manually crafted transformation rules. Second, it exhibits more robust generalization to a broader spectrum of transformation patterns, including those not encountered during development, whereas the rule-based system is intrinsically constrained by its fixed, predefined transformation hierarchy.

Crucially, the multi-step framework mitigates the systematic over-prediction characteristic of single-step LLMs by imposing intermediate structural constraints, thereby yielding more accurate and calibrated classifications. This finding indicates that explicit structured reasoning—rather than model scale alone—is a key factor in aligning LLM outputs with formally specified error taxonomies.

5.1 Error Analysis

Misclassifications across all approaches can be attributed to three primary sources: structural mismatch, intent ambiguity, and output-format inconsistency.

For the **rule-based system**, errors arise when the

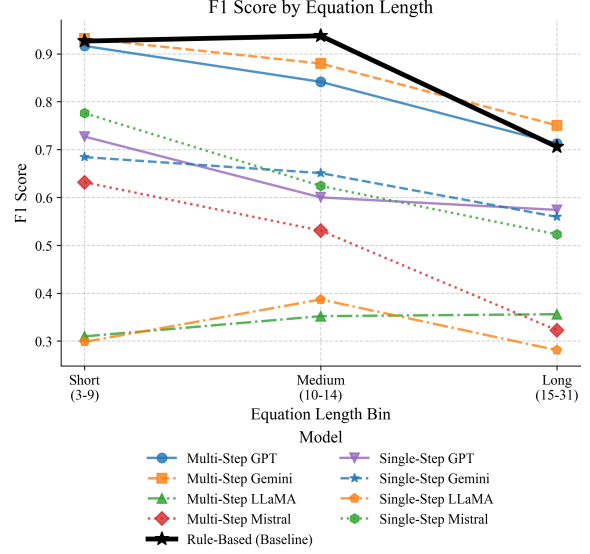


Figure 5: F1 scores across different equation length bins (Short, Medium, and Long).

similarity function fails to align a student’s transformation with any rule present in the transformation tree. For instance, consider the transition,

$$\begin{aligned} S_i : \quad -4t &= 1 - 5y \\ S_{i+1} : \quad t &= (5y + 1)/4 \end{aligned}$$

which is erroneously matched to,

$$best_match : y = (4t + 1)/5,$$

thereby resulting in an incorrect classification.

In the **multi-step architecture**, errors are primarily driven by difficulties in reliably inferring the student’s underlying intent, particularly when the observed transformations deviate substantially from expected intermediate forms or involve several implicit operations. For instance:

$$\begin{aligned} S_i : \quad z &= -10(x - 2) + 11(y + 1) - 21 \\ S_{i+1} \quad 10 &= -10x + 11y \end{aligned}$$

In this case, the model assigns a coarse-grained intent category (distribution and simplification) but fails to correctly classify the step due to the omission of the variable z and the substantial structural divergence between the two expressions.

For **single-step LLMs**, the dominant source of error is a discrepancy between the model’s internal reasoning and the strict operational definition of explicit sign errors. The models frequently produce correct mathematical interpretations but fail

to assign the corresponding label. For example:

$$\begin{array}{ll} S_i : & -2p - 2 = 10 \\ S_{i+1} : & p = 6 \end{array}$$

GPT response: *"No — explicit sign error: False. Reason: The correct solution is $p = -6$, and the student wrote $p = 6$. That is a wrong final sign but no incorrect sign operation is shown in the student's work."*

Although the model correctly infers that the mathematically valid solution is $p = -6$, it classifies the step as not containing an explicit sign error because no explicit sign manipulation is syntactically present in the student's work.

Open-Weight vs. Commercial Models: Open-weight models like Mistral and LLaMA often struggle to follow strict output formats, tending to produce free-form explanations instead of schema-compliant labels. This makes automatic parsing harder, complicates integration with downstream components, and can hurt overall pipeline performance. For instance, a typical LLaMA reply might say: *"The student's final simplified value has a sign error compared to the correct final value..."*. Although this is semantically correct, its unstructured form makes it difficult to automatically extract the underlying error category (e.g., a sign error). In contrast, commercial systems such as GPT and Gemini more reliably follow predefined schemas and generate structured, machine-interpretable outputs, which simplifies parsing and limits error propagation throughout the pipeline.

Impact of Equation Length: Equation length exerts a systematic negative effect on performance: all models do very well on short equations, begin to diverge on equations of medium length, and drop sharply on long equations (Figure 5). Rule-based methods and multi-step LLM approaches maintain relatively high accuracy on equations of medium length; however, all models struggle with long equations, where their performance converges and single-step LLMs are particularly impaired. These findings suggest that the increased compositional complexity associated with longer equations poses a specific challenge for single-step inference. Although multi-step reasoning strategies mitigate this effect to some extent, they do not fully resolve the underlying difficulty.

5.2 Summary

The results show that structured multi-step LLM systems can combine rule-based reasoning with generative flexibility to achieve higher precision and better performance by breaking tasks into sub-tasks and enforcing explicit structure on outputs. However, they still struggle with complex transformations, clarifying ambiguous student intent, and consistently following strict output formats, especially for open-weight models.

6 Conclusion and Future Work

In this study, we investigate the application of LLMs in various configurations for the detection of explicit sign errors in linear equation solving, with the objective of enabling accurate and timely feedback in large-scale educational environments. Using authentic student response data, we compare a carefully engineered rule-based baseline against zero-shot, few-shot, two-step, and three-step LLM-based pipelines, as well as hybrid variants augmented with CAS.

Our findings indicate that although the rule-based system constitutes a strong baseline, structured multi-step LLM approaches improve performance relative to single-step methods. In particular, three-step pipelines achieve the highest overall effectiveness, yielding a favorable balance between precision and recall and matching or surpassing the performance of the rule-based approach. In contrast, two-step pipelines exhibit instability, underscoring the importance of sufficient task decomposition. The integration of CAS results in only modest performance gains, suggesting that the primary challenges are rooted in reasoning and representation rather than in symbolic computation.

Future research will aim to generalize the proposed framework to encompass a wider spectrum of algebraic error categories and to address more structurally complex mathematical domains. Furthermore, we intend to systematically investigate hybrid architectures that more tightly integrate symbolic verification mechanisms with LLM-based inferential processes, with particular emphasis on enhancing the robustness and faithfulness of intermediate representations and mitigating error propagation across successive stages of the computational pipeline.

7 Limitations

This study is restricted to the detection of explicit algebraic sign errors occurring during transformations of linear equations, which limits the generalizability of our findings to other procedural errors commonly observed in student work, as well as to the treatment of more complex algebraic expressions that do not share the structural simplicity of linear equations. Extending the framework to additional error types would require corresponding modifications to the proposed multi-step architecture to handle a broader range of algebraic transformations. From a technical standpoint, we employed an identical prompt across all models while acknowledging that each model possesses distinct characteristics and prompt-sensitivity profiles. These model-specific properties may have influenced their respective outputs and, consequently, could have impacted the comparability of the results.

8 Ethics

This study utilizes screenshots of student solution steps, specifically those containing errors, collected from a digital mathematics learning platform. All data were fully anonymized prior to analysis, and no personally identifiable information was accessible to the authors at any stage of the study. All participating students provided informed consent via the platform's data usage agreement, permitting the use of their anonymized solution data for research purposes.

References

- John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.
- R. V. D. Bent, J. Jeuring, and B. Heeren. 2019. [The diagnosing behaviour of intelligent tutoring systems](#). pages 112–126.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- Joice Cazanowski Gomes and Patricia A Jaques. 2020. A data-driven approach for the identification of misconceptions in step-based tutoring systems. In *Simpósio*

Brasileiro de Informática na Educação (SBIE), pages 1122–1131. SBC.

- Harish Kirve, Prajakta Ghugare, Ayush Telrandhe, and Prajwal Mahale. 2025. [A fine-tuned transformer framework for the automated classification of algebraic misconceptions](#). *International Journal of Innovative Research in Technology*, 12(1):3648–3651.
- Hunter McNichols, Jaewook Lee, Stephen Fancsali, Steve Ritter, and Andrew Lan. 2024. Can large language models replicate its feedback on open-ended math questions? *arXiv preprint arXiv:2405.06414*.
- Hunter McNichols, Mengxue Zhang, and Andrew Lan. 2023. Algebra error classification with large language models. In *International Conference on Artificial Intelligence in Education*, pages 365–376. Springer.
- Thomas McTavish and Johann Larusson. 2014. Discovering and describing types of mathematical errors. In *Educational Data Mining 2014*.
- John W Ratcliff, David E Metzener, and 1 others. 1988. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, 13(7):46.
- Michael Russell, Laura M O'dwyer, and Helena Miranda. 2009. Diagnosing students' misconceptions in algebra: Results from an experimental pilot study. *Behavior research methods*, 41(2):414–424.

9 Appendix

9.1 Confusion Matrices

	Predicted Negative	Predicted Positive
Actual Negative	3104	153
Actual Positive	143	624

Table 3: Confusion matrix Rule-Based

	Predicted Negative	Predicted Positive
Actual Negative	3007	250
Actual Positive	438	329

Table 4: Confusion Matrix Mistral Model

	Predicted Negative	Predicted Positive
Actual Negative	2661	596
Actual Positive	21	746

Table 5: Confusion Matrix GPT-5-mini Model

	Predicted Negative	Predicted Positive
Actual Negative	3090	167
Actual Positive	619	148

Table 6: Confusion Matrix LLaMA 3 Model

	Predicted Negative	Predicted Positive
Actual Negative	2460	797
Actual Positive	11	756

Table 7: Confusion Matrix Gemini Model

	Predicted Negative	Predicted Positive
Actual Negative	2750	507
Actual Positive	355	412

Table 8: Confusion Matrix Multi-Agent LLaMA-3

	Predicted Negative	Predicted Positive
Actual Negative	2950	307
Actual Positive	365	402

Table 9: Confusion Matrix Multi-Agent Mistral

	Predicted Negative	Predicted Positive
Actual Negative	3210	36
Actual Positive	158	620

Table 10: Confusion Matrix Multi-Agent Gemini

	Predicted Negative	Predicted Positive
Actual Negative	3150	96
Actual Positive	179	599

Table 11: Confusion Matrix Multi-Agent GPT-5-mini

9.2 Prompt Examples

Prompt 1:

Detect whether there is an EXPLICIT sign error.
Explicit = wrong +/- is written.
Other = No Explicit sign error appears.
Original: {t0}
Student: {t1}
Explicit Sign Error: True/False

Prompt 2:

You are an expert at detecting sign errors in algebra.
A sign error is TRUE only if an incorrect sign (+ or -) is explicitly written.
Examples:
Original: $x + 5 = 15$
Student: $x = 15 + 5$
Explicit Sign Error: True
Original: $x + 5 = 15$
Student: $x = 20$
Explicit Sign Error: False
Now evaluate:
Original: {t0}
Student: {t1}
Explicit Sign Error: True or False

Prompt 3:

You are an expert at detecting sign errors in mathematical expressions.
Affirm whether there is a sign error in the student's work or not.
A sign error is TRUE only when the student's written work shows an incorrect sign for a variable or numeric (+ instead of -, or vice versa).

EXPLICIT vs IMPLICIT:

- EXPLICIT (TRUE): Original " $x + 1 = 2$ " Student writes " $x = 2 + 1$ " when it should be " $x = 2 - 1$ " → wrong sign is visible
→ sign error: TRUE
- IMPLICIT (FALSE): Original " $x + 1 = 2$ " Student writes → " $x = 3$ " when answer is 1 → might be sign error mentally, but no sign is shown → sign error: FALSE Example:

Original: $x + 5 = 15$
Student: $x = 15 + 5$
Explicit Sign Error: True
Implicit Sign Error: False

Original: $x + 5 = 15$
Student: $x = 20$
Explicit Sign Error: False
Implicit Sign Error: True

Original: $-(3x + 4) = 12$
Student: $-3x - 4 = 12$
Explicit Sign Error: False
Implicit Sign Error: False Does the following transformation contain an explicit sign error?

Original: t0
Student: t1

Prompt 4:

You are an expert at detecting sign errors in mathematical expressions. Affirm whether there is a sign error in the student's work or not. A sign error is TRUE only when the student's written work shows an incorrect sign for a variable or numeric (+ instead of -, or vice versa).

EXPLICIT vs IMPLICIT:

- EXPLICIT (TRUE): Original " $x + 1 = 2$ " Student writes " $x = 2 + 1$ " when it should be " $x = 2 - 1$ " → wrong sign is visible
→ sign error: TRUE

- IMPLICIT (FALSE): Original " $x + 1 = 2$ " Student writes → " $x = 3$ " when answer is 1 → might be sign error mentally, but no sign is shown → sign error: FALSE

Example:

Original: $x + 8 = 15$

Student: $x = 15 + 8$

Explicit Sign Error: True

Probability of an Explicit Sign Error: 100

Original: $-(3x - 4) = 12$

Student: $-3x - 4 = 12$

Sign Error: True

Probability of an Explicit Sign Error: 100

Original: $-(3x + 4) = 12$

Student: $-3x - 4 = 12$

Sign Error: False

Probability of an Explicit Sign Error: 1

Does the following transformation contain an explicit sign error?

Original: $\{t0\}$

Student: $\{t1\}$

9.3 Multi-Step Prompts

Step 1:

You are a math simplification agent. Your task is to mathematically simplify and canonicalize two equations.

ORIGINAL_EQUATION: "{original_eq}" STUDENT_EQUATION: "{student_eq}"

1. Combine all like terms (variables and constants) in the ORIGINAL_EQUATION to produce a canonical form.

2. Do the same for the STUDENT_EQUATION.

3. If the ORIGINAL_EQUATION can be fully solved, calculate the correct final value.

Respond with JSON in this exact format:

```
{ { "original_simplified": "canonical form of original", "student_simplified": "canonical form of student",  
  "original_final_value": "final value if solvable, or 'unsolvable'" }
```

Step 2:

You are an equation transformation agent. Your task is to infer the single logical step the student was attempting.

ORIGINAL_EQUATION: "{original_eq}"

STUDENT_EQUATION: "{student_eq}"

1. INFER INTENT: Describe the single major step (e.g., 'Transposing $3x$ to the RHS', 'Distributing -5 ').

2. PERFORM CORRECT TRANSFORMATION: Apply the inferred intent correctly to the ORIGINAL_EQUATION.

3. CHECK FOR PRE-EXISTING ERRORS: Compare canonical forms:

- Original Simplified: '{simplified_data.original_simplified}'

- Student Simplified: '{simplified_data.student_simplified}'

If the student's canonical form is radically different (indicating severe error like incorrect summation), set 'pre_existing_summation_error' to true.

Respond with JSON in this exact format:

```
{ { "student_intent": "description of intended operation",  
  "correct_intermediate_eq": "correct result of that operation",  
  "pre_existing_summation_error": true or false }
```

Step 3:

You are the final error classification agent. Isolate a single explicit sign error by comparing the correct transformation result with the student's actual result.

1. CORRECT INTERMEDIATE EQUATION (Target): "{correct_intermediate_eq}"
2. STUDENT'S RAW EQUATION: "{student_eq_raw}"
3. STUDENT'S SIMPLIFIED EQUATION: "{simplified_data.student_simplified}"
4. CORRECT FINAL VALUE: "{simplified_data.original_final_value}"
5. PRE-EXISTING SUMMATION ERROR: {transform_data.pre_existing_summation_error}

Classification Logic:

****Single Sign Flip Check:****

- Check if STUDENT'S RAW EQUATION could be made identical to CORRECT INTERMEDIATE EQUATION by flipping the sign of ONLY ONE TERM. Set 'is_single_flip_correction_possible' accordingly.

****Rule 1: Implicit/Summation Errors**** - IF PRE-EXISTING SUMMATION ERROR is True OR

is_single_flip_correction_possible is False:

- Set error_type='Summation_Error'

- Set is_error_in_intermediate_step=True

- Set error_found=False

****Rule 2: Explicit Sign Errors****

- ELSE (Single flip is True AND No pre-existing summation error):

- Set error_found=True

- Find the single sign-flipped term

- Determine if error was due to Transposition (moving term across '=') or Distribution (expanding parentheses)

- Set is_error_in_intermediate_step=False

****Rule 3: Arithmetic Simplification**** - If not Transposition/Distribution/Summation, check if student's final simplified value has a sign error compared to correct final value and classify as 'Arithmetic_Simplification'.

Respond with JSON in this exact format:

```
{ { "error_found": true or false,
  "error_type": "Transposition" or "Distribution" or "Arithmetic_Simplification" or "Summation_Error" or "None",
  "is_single_flip_correction_possible": true or false,
  "is_error_in_intermediate_step": true or false,
  "original_term": "term that was handled incorrectly",
  "correct_sign": "sign it should have",
  "student_sign": "sign student used",
  "reasoning": "explanation of classification"
}
```


Quality-Conditioned Agreement in Automated Short Answer Scoring: Mid-Range Degradation and the Impact of Task-Specific Adaptation

Abigail Victoria Gurin Schleifer¹ Moriah Ariely¹ Beata Beigman Klebanov²

Asaf Salman¹ Giora Alexandron¹

¹ Weizmann Institute of Science, Rehovot, Israel

² ETS, Princeton, USA

{abigail.gurin-schleifer, moriah.ariely, giora.alexandron, asaf.salman}

@weizmann.ac.il

bbeigmanklebanov@ets.org

Abstract

Automated short answer scoring (ASAS) is shifting from discriminative, fine-tuned models to large language models (LLMs) used in few-shot settings. This paradigm leverages LLMs' broad world knowledge and ease of deployment, but limited task-specific data may reduce alignment on complex scoring tasks. In particular, its impact on scoring partially correct responses that require nuanced interpretation remains underexplored. We investigate the relationship between the degree of task-specific adaptation of different models and quality-conditioned scoring agreement. We compare three LLMs (GPT-5.2, GPT-4o, Claude Opus 4.5) in few-shot mode, a fine-tuned BERT-based encoder, and a human expert on two open-ended biology items, using several hundred student responses and ground truth scores provided by a biology education expert. The results show that human-human agreement is highest and stable across the full quality spectrum. All AI models perform well on fully correct and fully incorrect responses, but exhibit substantial degradation on mid-range responses. This mid-range degradation is conditioned on task-specific adaptation: It is most severe in few-shot LLMs with few examples and decreases as task-specific data increases, with fine-tuned encoder models performing best. This mid-range degradation may lead to inequitable evaluation of responses produced by students with developing understanding. Our findings highlight the importance of quality-conditioned fairness, with particular attention to mid-range responses.

1 Introduction

Automated short answer scoring (ASAS) of open-ended responses is a central application of natural language processing (NLP) in education (Bonthu et al., 2021; Haller et al., 2022). ASAS approaches can be classified into reference-based – scoring based on similarity to reference graded responses –

and example-based, where the model learns to map student responses to rubric scores using task-specific labeled data. The fundamental trade-off is that reference-based approaches require only a few scored examples but their accuracy is typically very limited (Bexte et al., 2023), while example-based ones can reliably learn to mimic expert grading but require ample training data (Gurin Schleifer et al., 2023) and are still prone to gaming (Ding et al., 2020).

Early example-based systems relied on hand-crafted linguistic features and machine learning models to approximate human scoring (Haller et al., 2022). More recent work has leveraged neural machine learning that can capture deeper semantic relationships in student responses. Such neural-based automated assessment systems primarily relied on discriminative machine learning implemented on top of encoders that transferred raw responses into vectorized embeddings (Condor, 2020; Li et al., 2021; Gurin Schleifer et al., 2023).

The emergence of generative LLMs introduced a new paradigm for ASAS. Decoder-based language models can evaluate student responses based on their extensive pretraining and general world knowledge, relying on instructions to guide in-context interpretation (e.g., according to given rubrics), rather than on adapting their parameters using task-specific datasets (Lin et al., 2023).

However, because generative models are not explicitly trained on the application of a specific assessment framework to student responses, their scoring may be less aligned with expert grading, particularly in assessment contexts that have domain-specific standards and require subtle interpretation (Wu et al., 2025; Yacobson et al., 2025). Few-shot prompting, which is a prompting strategy that provides a few labeled examples, was developed to help the model align its interpretation with that of experts (Lin et al., 2023). This raises the question of the amount of task-specific data that

various types of architectures require to align with expert grading.

The alignment of automated scoring systems with human grading is typically evaluated using metrics like κ or Pearson's correlation between model predictions and human raters (Beigman Klebanov and Madnani, 2022; Williamson et al., 2012). That said, it is also recognized that overall scoring accuracy may not be sufficient for validating an automated scoring system, as its performance can vary systematically across student sub-populations, raising fairness concerns (Williamson et al., 2012). While the psychometric theory discusses sub-populations to be examined quite generally, e.g., as “identifiable and relevant” subgroups (Xi, 2010), much of the literature on fairness in educational AI tends to focus on demographics-based sub-groups (Loukina et al., 2019; Madaio et al., 2022) and on high-stakes, large-scale assessment (Johnson and McCaffrey, 2023). Discussing fairness in the formative classroom assessment specifically, Camilli (2006) noted that the use of the assessment is generally “to ‘locate’ the presenting proficiencies of students in order to guide instruction along the contours of their strengths and weaknesses”, and help instructors “see if there is anything that, if not addressed, sets certain students up to fail or otherwise miss significant opportunities to learn” (p. 248). To help provide the right learning opportunities for learners at different current states of knowledge, these states of knowledge need to be detected accurately and equitably, as otherwise students whose knowledge state tends to be mis-recognized would be in danger of being provided inappropriate learning support.

In the recent literature on automated analysis of student responses, there are indications that states of partial knowledge may be harder to detect automatically than completely correct knowledge states (Gurin Schleifer et al., 2025; Kortemeyer, 2023) or than both completely correct and completely incorrect ones (Grévisse, 2024). In such cases, students demonstrating partial mastery of a concept (mid-range responses) may receive less reliable evaluation than those whose responses are clearly correct or clearly incorrect. We refer as “quality-conditioned fairness” to the requirement that ASAS system performs equally well across the entire range of response quality.

Due to the nuanced interpretation required for mid-range responses, we expect that task-specific labeled data will be particularly important for AI

grading systems when grading such answers, and that its quantity will be associated with the systems' accuracy on these responses.

However, little is known about agreement patterns across the response quality spectrum, and how this relates to model architectures – encoder-based discriminative models and decoder-based generative ones – and across different levels of task-specific adaptation (few-shot with varying levels of ‘few’, and fine-tuning). Thus, our research is guided by the following research question:

RQ: *What is the association between response quality and inter-rater agreement for human-human vs human-AI pairings for different levels of task-specific adaptation of the AI models? (few-shot with different numbers of examples vs fine-tuned)?*

The data used for this study included responses from about 800 high-school students to two open questions in biology requiring students to provide scientific explanations in the domain of cellular respiration. The responses were graded according to an analytic grading rubric by two biology education experts. To answer the RQs, we compared three generative decoder-based models in a variety of few-shot modes, a discriminative fine-tuned encoder model and a human grader in terms of their agreement with the expert human grader overall and across different levels of response quality.

Previewing the results, we show that disagreements between human and AI raters concentrate among mid-quality responses, while agreement remains high at the extremes of the quality spectrum. They also demonstrate a consistent pattern of **mid-range degradation of AI scoring accuracy**, meaning that AI graders, but not the second human expert grader, tended to be less aligned with the expert on the mid-range responses, and that this tendency was partially mitigated as the AI system received more task-specific data.

These findings highlight the importance of task-specific adaptation for accurately evaluating partially correct responses and suggest that evaluation frameworks for AI-assisted assessment should explicitly consider quality-conditioned, and specifically mid-range, assessment fairness.

2 Related Work

Fine-tuning the parameters of the model on task-specific data is a common technique for adapting encoder-based pre-trained language models such

as BERT (Devlin et al., 2019) to the given task. The more recent models, while still utilizing the Transformer architecture, use orders of magnitude more parameters (hence their designation as *large* language models, or LLMs), and demonstrate in-context learning, namely, the ability to perform well on tasks that the model has not been trained for with only a small number of examples (few-shot). While larger models generally demonstrate stronger in-context learning, empirical studies suggest that parameters other than size, such as the design of the prompt, the model architecture, the diversity of the training data, and the selection of the few-shot examples can impact the model's in-context learning ability (Berti et al., 2025).

A number of studies compared the performance of fine-tuned encoders and few-shot LLMs on ASAS. Chamieh et al. (2024) evaluated ASAS performance on datasets from various domains, including mathematics, science, and English language arts. They found that fine-tuned models generally performed better, especially for tasks that require more complicated reasoning or domain-specific knowledge. Kortemeyer (2024) compared the performance of GPT-4, BERT, and RoBERTa-large (Liu et al., 2019) on ASAS datasets in varied science domains. They found that the fine-tuned BERT and RoBERTa-large outperform the general-purpose GPT-4 LLM. Henkel et al. (2024) found that few-shot models outperformed zero-shot ones in the context of ASAS, suggesting that more task-specific information may help improve scoring performance. Ferreira Mello et al. (2025) studied ASAS in English in the context of undergraduate computer science and in Brazilian Portuguese in 8th grade biology and found that fine-tuned BERT models outperformed few-shot GPT-4o on both datasets. Across these studies, the evaluation of performance was done using overall summary measures, such as RMSE, F1, Pearson's correlation, or kappa-family metrics. In particular, it remains unclear *what kinds of responses* were easier or harder for the different models to score.

There are indications in the literature on automated scoring of constructed responses that point towards a specific weakness of LLM-based scoring, namely, the tendency of these models to over-score low quality responses. Chang and Ginter (2024) examined the quality of GPT-4 scoring, in zero-shot and one-shot setting, of short answers in a variety of undergraduate courses in Finnish. They found that the models were too lenient, both in

binary pass/fail scoring and in scoring on a scale: Too few responses were marked as fail, and too few responses were assigned to the bottom half of the scale. Kortemeyer (2023) reported similar results for scoring short answers in introductory physics – GPT-4 was a more lenient rater. The author noted that agreement between human and AI scores was stronger at the high end of the response quality spectrum. In the context of medical undergraduate courses, Grévisse (2024) observed that different LLMs were more or less severe than human raters on data from different courses. They further observed that there was stronger agreement between the LLM and human ratings for the completely incorrect and completely correct answers than for partially correct answers and noted GPT-4's especially high precision on detecting the fully correct answers. These findings suggest that automated scoring should be examined from the point of view of response quality, as there is evidence that the scoring accuracy may be uneven across the score distribution. However, it is not clear whether few-shot LLM scoring is more or less susceptible to uneven performance than the more extensive task adaptation through fine-tuning. It is also not clear whether any unevenness in AI scoring may have to do with inherent difficulty of discriminating between scores at certain levels, namely, whether human scoring exhibits a similar pattern. Our contribution is a systematic examination of multiple scoring mechanisms – human, fine-tuned pre-trained language models, and few-shot LLMs – in terms of the relationship between scoring accuracy and response quality, in the context of ASAS.

3 Methodology

3.1 Assessment Items and Scoring Rubric

We analyzed student responses to two open-ended items in biology. The items deal with cellular respiration, and specifically, the effect of smoking and anemia on physical exercise. The items are conceptually similar and differ in context and surface features, and students' explanations are expected to have the same structure, and use similar argumentative flow. They are similar to questions that frequently appear on the biology national matriculation exam. They were administered in Hebrew and their translation is presented below.

Smoking (Item 1): Cigarette smoke contains harmful substances, including CO, which binds haemoglobin more strongly than oxygen. Explain

how high CO levels impair exercise ability.

Anemia (Item 2): A man with low red blood cell levels (anaemia) reports weakness and difficulty exercising. Explain how reduced red blood cells make exercise difficult.

Due to their conceptual similarity, the items share the same grading rubric. It is an analytic grading rubric that decomposes a student’s scientific explanation into a collection of 10 binary categories, each representing an essential property that the student response should include. The items and the rubric were developed by the 2nd author of this paper. The full description of the grading rubric can be found in [Ariely et al. \(2025\)](#).

3.1.1 Data Collection, Scoring, and Partitioning

Student responses were collected in two cycles conducted in two different academic years, with a two-year gap between the cycles. In the first cycle, student responses to an instrument that contained both items were collected from 669 students in grades 10-12 attending 25 high schools across the country. For each category in the rubric, a student’s response was graded as ‘1’ if that category was addressed in the text, and ‘0’ otherwise. Approximately 5% of the 669 responses were annotated jointly by two human raters – Rater1 (the second author) and Rater2. Then, an additional 20% of the 669 responses were annotated independently by both raters establishing inter-rater agreement on $n = 130$ responses (κ ranged between 0.89 and 0.98 across categories). Last, Rater1 coded the remaining responses; see [Ariely et al. \(2023\)](#) for more details on the annotation process. The data was used as **training data** for the models, namely, for fine-tuning the BERT-based models and for picking the few-shot examples for the LLM-based grading.

In the second cycle, responses to the same instrument were collected from 152 students (in total, 304 responses) of similar demographics, and scored by Rater1 (the 2nd author of this paper). We refer to this scoring as the ground truth labels. These responses were used as **test data** for all the experiments described below. A second human rating was obtained for the test items as part of the current study of human vs AI raters and will be described in more detail in section 3.3.2.

3.2 Evaluation Metric for Scoring Differences

The scoring of each response is represented as a 10-dimensional binary vector, where each dimension corresponds to a category, with ‘1’ indicating a correct response on that category. As noted above, we referred to the scoring of Rater1 as the ground truth labels. The overall score of a response is thus defined as the number of categories (i.e., bins) that Rater1 marked as correct, which ranges from 0 to 10, meaning that higher scores indicate higher response quality.

3.2.1 Manhattan Distance for Scoring Differences

For a student response s , we denote the gold labels of s by: $l_s^G = (l_1^G, \dots, l_{10}^G)$, and the labels assigned by Rater2 (Human or AI-based) – by: $l_s^{R_2} = (l_1^{R_2}, \dots, l_{10}^{R_2})$. l_s^G and $l_s^{R_2}$ are binary 10-dimensional vectors. We measure the absolute difference between the two scorings by the Manhattan-distance (also called the L_1 -distance) ([Chen and Ng, 2004](#)) between l_s^G and $l_s^{R_2}$:

$$\|l_s^G - l_s^{R_2}\|_1 = |l_1^G - l_1^{R_2}| + \dots + |l_{10}^G - l_{10}^{R_2}|, \quad (1)$$

which equals the number of categories scored differently by Rater2 compared to the gold labels.

3.3 Experimental Set-Up

3.3.1 The Raters and the Scoring Task

The experimental set-up included comparing the alignment to the ground truth labels of the following scoring mechanisms: (i) a second human expert, (ii) a fine-tuned BERT-based model, and (iii) three LLMs – GPT-5.2 ([Singh et al., 2025](#)), GPT-4o ([OpenAI, 2023](#)), and Claude-opus-4.5 ([Anthropic, 2025](#)). At the time of writing, GPT-5.2 and Claude-opus-4.5 are the frontier models of OpenAI and Anthropic, while GPT-4o represents OpenAI’s previous generation model, which is the current workhorse for many LLM tasks due to its good combination of capabilities, speed, and cost.

The task of the graders was to score each response of the test set (152 responses for each item) according to each of the categories of the rubric, producing a 10-dimensional binary vector per response. For the LLMs we took the majority result: Each LLM was invoked three times for every category on every response; the result that appeared most of the time is considered the final LLM score.

We then computed the Manhattan distance, per response and rater, between the Rater2’s scoring

vector and the ground truth vector, as described in Section 3.2.1. Below, we provide more details about the scoring procedure.

3.3.2 Scoring Procedures by Scoring Mechanism

Below we describe how the scoring of the test set was implemented for each scoring mechanism.

1. **Human:** The test set was labeled according to the rubric by a second biology education expert who scored part of the 1st cycle dataset (that was used as training data in this study) a few years ago, after a short recap with Rater1 on the rubric and its implementation.
2. **Fine-tuned BERT-based Classifier:** The BERT-based model DictaBERT (Shmidman et al., 2023) was fine-tuned on the training data ($n = 669$). A separate binary classifier was trained for each item and category, meaning that overall, 20 classifiers were fitted. More details can be found in Ariely et al. (2025).
3. **Few-shot LLMs:** The generative models – GPT-4o, GPT-5.2, and Claude Opus 4.5 – were prompted with assessment instructions. Each category in the rubric had its own prompt for each item, overall $2 \times 10 = 20$ prompts. The prompts followed a straightforward few-shot ASAS strategy and their general structure was: role definition, task definition, per-category scoring instructions, and few-shot examples, split equally between positive and negative examples. An example of a prompt for one of the items and categories is provided in Appendix A. We acknowledge that there are various prompting strategies for ASAS. Our approach was to rely on established prompting strategy developed and tested in previous ASAS research. Thus, we adopted the prompts of Ariely et al. (2025), which were initially developed for a GPT-4o grader. We note that Ariely et al. (2025) tried several combinations of few-shot examples, specifically 2/4/6/8/10-shot (with an equal split between positive and negative examples). **Determining the Number of Few-Shot Examples for the Frontier Models.** Prior to experimenting with the costly models (GPT-5.2 and Claude Opus 4.5), we analyzed the results of running the 2/4/6/8/10-shot prompts with GPT-4o. The results showed that 10-shot prompting achieved better results, both in terms of overall (average) agreement and in terms of mid-range fairness, as can be seen in Figure 1.

Thus, for the costly models, we ran only the 10-shot version. For the few-shot examples, we randomly chose five positive and five negative examples from the training set. The total cost of running the ten 10-shot prompts on the test set for each item for each of the two LLMs (40 prompts in total) was around \$1,000.

4 Results

Table 1 shows the average L1 distance between the gold labels and the scoring of the second human rater (H2) and the AI models, per score level (number of correct categories – #CC – in the gold labels), for each of the two items. Figure 1 shows the same information visually, including standard deviations shown in the relevant background color.

Let us first consider the overall performance of the different models on the two items. On average and across score levels, H2 had the highest agreement with the expert, followed by the fine-tuned BERT-based models, followed by the LLMs in few-shot mode. Among the GPT models, we did not observe an advantage for GPT5.2 over the prior generation GPT4o model with the same number of few-shot examples (10). For GPT4o – the only model that was tested with several numbers of few-shot examples – providing 6-10 examples resulted in approximately similar performance, while providing only four examples resulted in performance degradation on Item 2, and providing only two examples resulted in a substantial degradation in the middle range of the scores – difference of 3 categories or more, on average, for scores with 4-6 correct categories out of 10 – for both items.

Analyzing the results by score level (number of correct categories out of 10), we observe excellent AI-model performance on the extremes – best and worst quality responses – regardless of model-type and amount of task-specific data. In contrast, there is a degradation for partially correct responses, especially for the few-shot models.

While on average fine-tuned models performed better than few-shot ones, the GPT models in all few-shot scenarios – including those with only two examples – performed very well on fully or almost fully correct answers (score 9-10), showing average distance of less than one category from the gold score. On the fully correct responses, the GPT models outperformed the fine-tuned models on both items. The exceptionally strong performance of few-shot models on the best responses

aligns with some similar reports in the literature (e.g., Grévisse (2024); Kortemeyer (2023)).

Next, we observe that all GPT models perform well on the lowest end of the scale – score levels 0 and 1. With the exception of two-shot GPT4o on Item 2, in all other cases the average distance from the gold label is less than one category. Although fine-tuned models do better, with average distance of at most 0.5 category from the gold label for score 0 and 1 across the two items, few-shot models are also fairly accurate.

Table 1 shows that all few-shot models suffer a substantial degradation in performance in the middle range of the scores. For all six GPT models, for each of the four score levels 4-7, for both items, all but one of the $6 \times 4 \times 2 = 48$ results show average distance of more than two categories from the gold label. Fine-tuned models also perform worse in this middle range than in the extremes, but the degradation is not as severe – the maximum average distances from the gold scores is 1.5 categories. In contrast, the human rater H2 does not show a clear pattern of degradation in agreement with the gold labels in the middle of the score range.

Our final observation is that of the erratic behavior of the Claude model. While the overall and the score-level-dependent patterns of errors are highly consistent across the two items for H2, fine-tuned, and GPT family models, Claude’s performance oscillates dramatically – it is the best few-shot model for Item 1 but is by far the worst model for Item 2.

5 Discussion & Conclusions

Quality-conditioned agreement. The findings show that agreement is strongly conditioned by response quality and that this relationship differs considerably between human-human and human-AI pairings. Human raters exhibit the highest agreement, which is also stable across the response quality range and shows no mid-range degradation, suggesting that human experts are relatively consistent in interpreting partially correct responses. In contrast, the AI models showed a U-shaped pattern:¹ Agreement is highest at the extremes (fully correct or incorrect responses) and substantially lower in the middle range, indicating that model-human agreement is not uniform across the scoring range. This mid-range degradation pattern complements a recent ASAS study showing that criterion-level

agreement declines as cognitive complexity increases (Emirtekin and Özarslan, 2026), and a similar trend found in project-based learning (Usher and Faraon, 2025).

Task Specific Adaptation. The results also showed that task-specific adaptation emerged as a key factor in improving agreement. The fine-tuned model, which had access to the highest amount of task specific data, outperformed scoring based on LLMs with few-shot examples across both items, most notably on mid-range responses. In line with this, the initial experiment on the number of few-shot examples on GPT-4o demonstrated a similar pattern, with the U-shape disagreement curve having a larger ‘belly’ for lower numbers of few-shot examples (see Figure 1). While the advantage of fine-tuned models on LLMs in few-shot mode on ASAS tasks was shown previously on the aggregated level (Kortemeyer, 2024; Henkel et al., 2024; Ferreira Mello et al., 2025), the present work suggests that this advantage may be mostly concentrated in the mid-range examples. In fact, for the fully correct/incorrect responses, LLMs performed well with very little task-specific data.

Possible explanations for the observed pattern. In trying to explain these results, it may be that extreme responses (fully correct or incorrect) provide clearer signals that are easier to classify, while mid-quality responses require nuanced judgment and interpretation of rubric criteria. Experienced human experts may be better at applying such nuanced educational judgment (Yacobson et al., 2025), and encoder models can learn to distinguish between them using labeled data that changes their internal representation in a way that can separate between them. LLMs, when provided with a rubric and few-shot examples, utilize in-context pattern matching to identify semantic and structural alignments between the criteria, the examples, and the student’s input. Our empirical results that show that LLMs are less good than fine-tuned encoders on ASAS tasks are in line with previous findings (e.g., Ferreira Mello et al. (2025)).

Implication for automated scoring in practice. The great promise of few-shot automated scoring is fast deployment of the automated scoring model, thus saving instructors valuable time through automated evaluation of student responses without having to wait until sufficient human-scored data is available for fine-tuning. Our results suggest that such scoring may be inequitable, with substantially inaccurate scores concentrating on the partially in-

¹Specifically, the graphs in Figure 1 show an inverted U-shape since they show discrepancy rather than agreement.

#CC	H2	FT	Claude	GPT 5.2	GPT 4o10	GPT 4o8	GPT 4o6	GPT 4o4	GPT 4o2	Me- dian
Item 1										
0	0	0.09	0.80	1.10	0.60	0.70	0.80	0.90	0.80	0.80
1	0.12	0.25	0.40	0.90	0.40	0.60	0.50	0.70	0.90	0.50
2	0.08	0.23	0.64	1.32	0.64	0.64	0.86	1.05	1.95	0.64
3	0.67	1.33	1.27	2.45	2.09	1.55	1.91	2.18	3.18	1.91
4	0.45	1.27	1.33	2.25	2.17	2.83	2.67	2.83	3.17	2.25
5	0.12	1.12	1.30	2.70	2.4	3.00	2.60	2.60	3.50	2.60
6	0.43	0.91	1.61	2.96	2.43	2.91	2.64	2.32	3.00	2.43
7	0.12	1.44	1.22	2.44	2.33	2.56	2.22	1.83	2.61	2.22
8	0.12	0.56	0.88	1.38	1.38	1.62	1.44	1.25	1.25	1.25
9	0	1.17	0.50	0.67	0.83	0.83	0.83	0.67	0.33	0.67
10	0.09	0.27	0.09	0.09	0.18	0.18	0.18	0.18	0.09	0.18
Avg.	0.20	0.79	0.91	1.66	1.40	1.58	1.51	1.50	1.89	
Stdev.	0.42	0.86	0.82	0.98	0.79	0.91	0.96	0.95	0.93	
Item 2										
0	0	0.17	5.57	0.71	0.29	0.57	0.43	0.86	0.86	0.57
1	0.38	0.50	4.90	0.90	0.70	0.60	0.60	0.90	1.50	0.70
2	0.50	0.75	5.14	0.57	0.57	1.00	0.71	1.29	2.29	0.75
3	0.45	0.64	4.38	1.43	0.71	0.93	1.21	1.57	2.43	1.21
4	0.14	0.86	4.25	2.50	2.62	2.62	2.38	3.00	3.00	2.62
5	0.33	1.50	4.00	2.33	2.22	2.44	2.33	2.33	3.33	2.33
6	0.50	1.29	4.06	2.61	2.78	2.83	3.00	3.28	3.28	2.83
7	0.11	1.39	3.50	2.59	2.55	2.64	2.68	2.82	2.86	2.64
8	0.42	1.23	2.70	1.32	1.82	1.86	1.89	1.75	1.79	1.79
9	0.25	1.12	1.76	0.71	0.88	0.76	0.94	0.82	0.82	0.82
10	0.09	1.18	2.18	0	0	0	0	0	0	0
Avg.	0.29	0.88	3.86	1.42	1.38	1.48	1.47	1.69	2.01	
Stdev.	0.49	0.85	1.99	0.90	0.94	0.86	0.90	0.97	1.09	

Table 1: Average L1 distance per number of correct categories (#CC), based on the gold labels. H2 is the second human rater. Top panel: Item 1; bottom panel: Item 2. Light gray cells show L1 distance ≥ 1 ; darker gray shows L1 distance ≥ 2 ; the darkest gray shows L1 distance ≥ 3 .

correct responses. However, the high agreement on fully correct and fully incorrect responses, even for LLM models with very few examples, suggests that automated scoring as a first pass can identify these responses and handle them without human intervention, while responses predicted to be in the middle range would be routed to the human rater for scoring. Gradually, as more human-scored data is accumulated and fine-tuning becomes feasible, the scoring can be automated more fully.

Implications for fairness. Bias is a well-recognized hazard in automated scoring of constructed responses (Madnani et al., 2017; Loukina et al., 2019; Gorgun and Yildirim-Erbasli, 2026). Much of the existing literature has focused on demographic-group-based biases, where students

from different groups (e.g., by socioeconomic status (Chinta et al., 2024; Baker and Hawn, 2022) gender (Madnani et al., 2017), or linguistic background (Loukina et al., 2019)) receive systematically different scores despite comparable skill levels. However, bias can also arise at the individual level, where similarly skilled students are treated differently due to consistent features of their responses – for example, students who write in a more concise style may be systematically scored differently (Madnani et al., 2017; Beigman Klebanov and Madnani, 2022). In addition, statistical biases may emerge from properties of the data, for instance, if examples that include certain properties, such as specific misconceptions, are not represented enough in the data for the models to learn

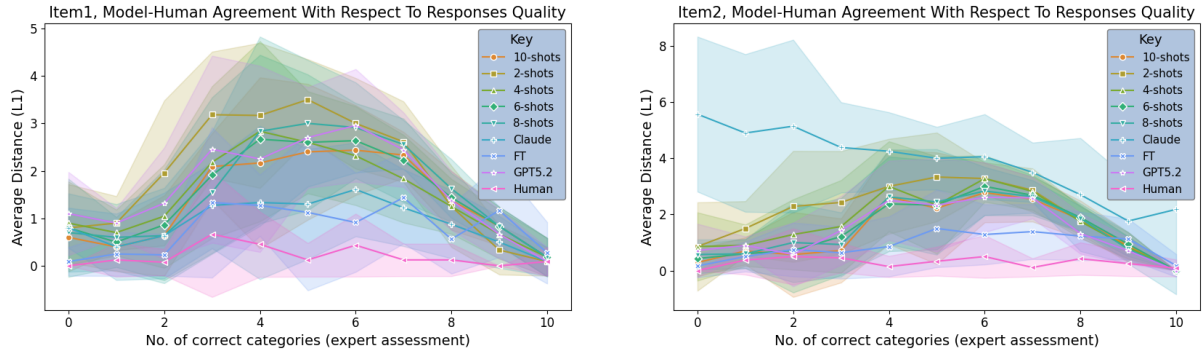


Figure 1: Human-Model Agreement With Respect To Student's Response Quality For All The Models

their linguistic signature.

Statistical bias can also take the form of a representation bias. For example, Gurin Schleifer et al. (2025) identified what they termed an ‘Anna Karenina principle’, showing that as responses become more incorrect, their representations in the embedding space become more spread, making it harder to automatically cluster together responses of the same error type. This creates a differential measurement error that disproportionately affects lower-quality responses, and may constitute a fairness problem if such clusters are used to design feedback targeted at each knowledge group, as proposed by Ariely et al. (2024). In this case, low-quality responses may receive unsuitable feedback.

Our results contribute to this line of research by identifying a form of measurement bias that appears in ASAS. We observe a U-shaped pattern in which scoring errors concentrate in mid-range responses. These responses are typically produced by students whose knowledge is still developing. Such students may benefit the most from formative feedback, as they are often better able to use feedback to refine their understanding than lower-performing students (Roll et al., 2014; Shute, 2008). Additionally, because mid-range responses contain both correct and incorrect elements, scoring errors – especially misjudging correct components – may be particularly confusing. As a result, reduced accuracy in this range may hinder learning at a critical stage, making this a fairness concern.

We define quality-conditioned fairness as the requirement that automated scoring systems maintain consistent accuracy across the full spectrum of response quality. As we showed, this form of fairness is closely tied to the availability of task-specific data, with greater adaptation reducing mid-range errors. By identifying and characterizing this

pattern, we turn it from an “unknown bias” into a “known bias,” which is a necessary step toward improving fairness in educational AI systems in general (Baker and Hawn, 2022).

Future work. To better understand the phenomenon of mid-range degradation, it would be useful to extend this research to additional instruments and domains, and to examine the direction of misalignment, as prior work suggests that LLMs may tend to over-score incorrect responses. To improve mid-range alignment, it would be valuable to investigate whether including more mid-quality examples in few-shot prompts can reduce the observed degradation in scoring. A more human-in-the-loop approach worth investigating is whether task-specific prompts can achieve higher alignment, and how to train domain experts or teachers to develop and validate them within their local context.

Contribution. This work identifies mid-range scoring degradation in ASAS and shows that it is strongly governed by the degree of task-specific adaptation. A key takeaway is that LLM-based ASAS systems must be evaluated for quality-conditioned agreement, with particular attention to mid-range responses.

Limitations

The prompting approach used for the LLM-based scoring mechanisms followed a common few-shot strategy drawn from prior work. However, these prompts were originally designed and validated for GPT-4o. It is possible that alternative prompting strategies, or prompts tailored to each specific model, could yield different performance patterns. Additionally, the assessment used in this study consisted of two items in biology. Although these items were selected to represent common question types found in standard assessments on the topic,

they still constitute a limited sample of possible tasks. Thus, the generalizability of the findings to other items and domains remains to be established.

This research does not employ standard evaluation metrics such as Quadratic Weighted Kappa (QWK) or Pearson’s correlation, which may hinder direct comparison with existing ASAS benchmarks. Incorporating standard quality-conditioned agreement metrics should be considered in future work.

Ethics statement

We acknowledge that this work complies 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.

Acknowledgments

The authors thank Cipy Hofman for her contribution to the research. This work was supported by the Knell Family Institute for Artificial Intelligence, Israel.

References

- Anthropic. 2025. [Claude opus 4.5 system card](#). Technical report, Anthropic. System card.
- 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*, 61(8):1858–1889.
- Moriah Ariely, Asaf Salman, Anat Yarden, and Giora Alexandron. 2025. Reflective prompt engineering: a new strategy for automated short answer scoring in biology. *International Journal of Science Education*, pages 1–23.
- Ryan Baker and Aaron Hawn. 2022. Algorithmic bias in education. *International journal of artificial intelligence in education*, 32(4):1052–1092.
- Beata Beigman Klebanov and Nitin Madnani. 2022. *Automated essay scoring*. Springer Nature.
- Leonardo Berti, Flavio Giorgi, and Gjergji Kasneci. 2025. Emergent abilities in large language models: A survey. *arXiv preprint arXiv:2503.05788*.
- 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.
- Sridevi Bonthu, S Rama Sree, and MHM Krishna Prasad. 2021. Automated short answer grading using deep learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pages 61–78. Springer.
- Gregory Camilli. 2006. Test fairness. *Educational measurement*, 4:221–256.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- Li-Hsin Chang and Filip Ginter. 2024. Automatic short answer grading for finnish with chatgpt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23173–23181.
- Lei Chen and Raymond Ng. 2004. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803.
- Sribala Vidyadhari Chinta, Zichong Wang, Zhipeng Yin, Nhat Hoang, Matthew Gonzalez, T Le Quy, and Wenbin Zhang. 2024. Fairaid: Navigating fairness, bias, and ethics in educational AI applications. *arXiv preprint arXiv:2407.18745*.
- Aubrey Condor. 2020. Exploring automatic short answer grading as a tool to assist in human rating. In *International Conference on Artificial Intelligence in Education*, pages 74–79. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yuning Ding, Brian Riordan, Andrea Horbach, Aoife Cahill, and Torsten Zesch. 2020. [Don’t take “nswvt-nvakgxp” for an answer –the surprising vulnerability of automatic content scoring systems to adversarial input](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 882–892, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Emrah Emirtekin and Yasin Özarslan. 2026. Automatic short-answer grading in sustainability education: AI-human agreement. *Journal of Computer Assisted Learning*, 42(1):e70160.

- Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. 2025. [Automatic short answer grading in the LLM era: Does GPT-4 with prompt engineering beat traditional models?](#) In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, LAK '25, page 93–103. Association for Computing Machinery.
- Guher Gorgun and Seyma N. Yildirim-Erbasli. 2026. [Algorithmic bias in BERT for response accuracy prediction: A case study for investigating population validity](#). *Journal of Educational Measurement*, 63(1):e12420.
- Christian Grévisse. 2024. LLM-based automatic short answer grading in undergraduate medical education. *BMC Medical Education*, 24(1):1060.
- Abigail Gurin Schleifer, Beata Beigman Klebanov, and Giora Alexandron. 2025. Uncovering measurement biases in LLM embedding spaces: The anna karenina principle and its implications for automated feedback. *International Journal of Artificial Intelligence in Education*, 35(5):2821–2855.
- 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 (BEA 2023)*, pages 550–555.
- Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. 2022. Survey on automated short answer grading with deep learning: from word embeddings to transformers. *arXiv preprint arXiv:2204.03503*.
- Owen Henkel, Libby Hills, Adam Boxer, Bill Roberts, and Zach Levonian. 2024. [Can large language models make the grade? an empirical study evaluating LLMs ability to mark short answer questions in k-12 education](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, L@S '24, page 300–304, New York, NY, USA. Association for Computing Machinery.
- Matthew S Johnson and Daniel F McCaffrey. 2023. Evaluating fairness of automated scoring in educational measurement. In *Advancing natural language processing in educational assessment*, pages 142–164. Routledge.
- Gerd Kortemeyer. 2023. [Toward AI grading of student problem solutions in introductory physics: A feasibility study](#). *Phys. Rev. Phys. Educ. Res.*, 19:020163.
- Gerd Kortemeyer. 2024. Performance of the pre-trained large language model gpt-4 on automated short answer grading. *Discover Artificial Intelligence*, 4(1):47.
- Zhaohui Li, Yajur Tomar, and Rebecca J Passonneau. 2021. A semantic feature-wise transformation relation network for automatic short answer grading. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6030–6040.
- Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*.
- 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*.
- 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.
- Michael Madaio, Su Lin Blodgett, Elijah Mayfield, and Ezekiel Dixon-Román. 2022. Beyond “fairness”: Structural (in) justice lenses on AI for education. In *The ethics of artificial intelligence in education*, pages 203–239. Routledge.
- Nitin Madnani, Anastassia Loukina, Alina von Davier, Jill Burstein, and Aoife Cahill. 2017. [Building better open-source tools to support fairness in automated scoring](#). In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 41–52, Valencia, Spain. Association for Computational Linguistics.
- OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Ido Roll, Ryan SJ d Baker, Vincent Aleven, and Kenneth R Koedinger. 2014. On the benefits of seeking (and avoiding) help in online problem-solving environments. *Journal of the Learning Sciences*, 23(4):537–560.
- Shaltiel Shmidman, Avi Shmidman, and Moshe Koppel. 2023. Dictabert: A state-of-the-art bert suite for modern hebrew. *arXiv preprint arXiv:2308.16687*.
- Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research*, 78(1):153–189.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. OpenAI gpt-5 system card. *arXiv preprint arXiv:2601.03267*.
- Maya Usher and Montathar Faraon. 2025. Who grades best? comparing chatgpt, peer, and instructor evaluations across varying levels of student project quality. *Assessment & Evaluation in Higher Education*, pages 1–20.

- 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.
- X. Wu, P. P. Saraf, G. Lee, and 1 others. 2025. [Unveiling scoring processes: Dissecting the differences between LLMs and human graders in automatic scoring](#). *Technology, Knowledge and Learning*.
- Xiaoming Xi. 2010. How do we go about investigating test fairness? *Language testing*, 27(2):147–170.
- E. Yacobson, S. Rapp, R. Blonder, and G. Alexandron. 2025. [Human experts vs. LLMs: Who is better at explaining student clustering?](#) In *Proceedings of the 2nd Human-Centric eXplainable AI in Education (HEXED) Workshop at EDM 2025*.

A Prompt Example

The general prompt structure is taken from [Ariely et al. \(2025\)](#). The scoring instructions are tuned per category. Each category-specific prompt is then followed by randomly chosen few-shot positive and negative examples.

1. Role definition:

'You are an expert in education and assessment in biology.'

2. Task definition and general instructions:

'You are required to assign a score to category in a student's response titled: "Changes in the rate or amount of energy Production." Note that although the overall context of the response is important, you should focus solely on the students' reference to changes in the rate or amount of energy or ATP production. Anything else mentioned around this category may be significant for the response, but irrelevant for evaluating this specific category.'

3. Instructions for scoring:

'If the category is present in the response, score 1. If not, score 0.'

Using LLMs for Item Creation: Validating the Potential of Automatically Generated Sentence Repetition Test Items for Language Assessment

Sarah Löber^{1,2}, Björn Rudzewitz^{1,2}, Yuan Chu³, Mengyuan He³, Shiqin Liu³,
Yushan Ye³, Xiaobin Chen^{1,2}

¹Hector Research Institute of Education Sciences and Psychology,
University of Tübingen, Germany

²LEAD Graduate School and Research Network, University of Tübingen, Germany

³Guangzhou City University of Technology, China

{sarah.loeber,bjoern.rudzewitz,xiaobin.chen}@uni-tuebingen.de
{chuyuan,hemy,liusq,yeys}@gcu.edu.cn

Abstract

Various aspects of the Elicited Imitation Test (EIT), a sentence repetition task for language assessment, can be automated, for example in terms of test administration or automatic scoring. It is potentially also possible to generate test items with Large Language Models (LLMs). This study investigates the potential of GPT-4o for item creation in the context of EIT, creating a parallel form to two popular and validated tests. We analysed the tests in terms of their linguistic and psychometric properties. While the items created by the LLM show some difference in grammatical structures when compared to human-written items, linguistic complexity results did not differ significantly between tests. Psychometric properties showed only minor differences. These findings lend support to the potential of Automatic Item Generation with LLMs in the context of sentence repetition tasks and might support the process of standardisation in SLA research and testing by enabling parallel test creation.

1 Introduction

Creating items for language tests places a particular kind of strain on the test developer: it requires resources such as personnel and time as well as expertise, making it a complex and labourious process. In the realm of language assessment, recent advancements have made possible the automation of this task. For example, [Settles et al. \(2020\)](#) used machine learning techniques for the automatic creation of proficiency assessments for adaptive testing. Likewise, [Attali et al. \(2022\)](#) utilised a transformers-based approach for generating reading comprehension assessments. As these examples make apparent, generating items automatically can accelerate the process of test creation and reduce the amount of labour typically associated

with this task. Indeed, Automatic Item Generation (AIG) can be found in many contexts nowadays, from generating reading comprehension questions ([Huang and He, 2016](#); [Sayin and Gierl, 2024](#)) to multiple choice items ([Gierl et al., 2012](#)). Another test format that lends itself well to automation and therefore AIG is the Elicited Imitation Test (EIT). While AIG is a promising and efficient approach, questions remain about construct representation and validity.

Traditionally, item creation for the EIT has been manual, relying on test creators for writing their own items and tests. Since EIT is a sentence repetition task, the sentences need to be carefully controlled in terms of length and linguistic complexity. Automating this process would reduce the amount of work related to item creation and increase the possibility of adaptive testing in the context of EIT. Furthermore, AIG has the potential to tackle another problem: the lack of standardisation in EITs and in language research generally ([Isbell and Son, 2022](#)), which becomes problematic when researchers want to compare a construct, e.g. language proficiency, across studies or when the need for repeated assessment arises. Parallel and comparable forms are a solution to this issue, provided they demonstrate comparable psychometric properties.

As suggested in recent literature, pre-trained LLMs and prompting are increasingly used for engineering test items ([Song et al., 2025](#)). One clear advantage of this approach is that using pre-trained LLMs is an accessible option for most people without a strong computational background that is otherwise needed to engineer systems for automatic item generation. However, questions remain about the quality and linguistic properties of these generated items, particularly in the context of validity in

(language) testing.

The aims of this study are to (1) investigate the linguistic properties of automatically generated items with LLMs in the context of a parallel form of an EIT, and (2) to examine if these test items display validity evidence in order to support the development of standardised and comparable EIT forms.

2 Theoretical background

2.1 Elicited Imitation

The Elicited Imitation Test (EIT) is a sentence repetition task, commonly used as a measure of general language proficiency. The test is also used in Second Language Acquisition (SLA) research, for example, to measure implicit grammatical knowledge of a second language (Erlam, 2006). Furthermore, the EIT can also be used as a placement test (Yan et al., 2020) or as a diagnostic tool for language acquisition in children (Klem et al., 2015).

The procedure of an EIT is as follows: The test taker hears sentences one by one, which they have to repeat verbatim. There is usually a short pause between the stimulus and the repetition to avoid over-reliance on working memory. The rationale behind sentence repetition is that the test taker reconstructs the meaning of the sentence based on their own grammatical knowledge.

Various design implementations can be employed when designing an EIT, for example using sentences of varying length (Ortega et al., 2002), including a distractor instead of a pause (Erlam, 2006), including ungrammatical sentences, or including only items that target a specific construct such as the passive-form (Spada et al., 2015).

So far, most of the test creation in the context of the EIT has been manual. Items have been written by researchers or test developers, which is a labour-intensive and complex task. Furthermore, it makes scaling of the test for parallel testing or testing on a larger scale difficult. This can cause problems as comparable testing and standardisation are two things that are currently lacking in SLA research (Isbell and Son, 2022). This difficulty could be tackled by automating EI test generation and administration to increase its scalability.

Research efforts have concentrated on automating the EIT using various technologies (McGuire and Larson-Hall, 2025; Isbell et al., 2023; Kim et al., 2024). For instance, Isbell et al. (2023) used Automatic Speech Recognition (ASR) and string

matching to automate the scoring of the Korean EIT, achieving very high correlations ($> .90$) between human and automated scores. Automatic scoring has also been implemented for written EITs (Chiffigarov et al., 2025). Furthermore, the EIT can be digitised and taken on the web, yielding results highly similar to lab-based approaches (Kim et al., 2024). In addition to these two areas, researchers have also investigated item creation for EITs (Christensen et al., 2010). Unlike delivery and scoring, EIT item creation has largely been manual.

2.2 Item Creation Approaches in EIT

Items used in existing EITs have traditionally been hand-crafted. For example, the most well-known EIT by Ortega et al. (2002) used hand-crafted items, as do the parallel forms in several languages. In order to create parallel English versions for comparability, Wu et al. (2022) also hand-crafted items based on the original Ortega EIT. Using appropriate items in the context of EIT is important, since the purpose of the test can vary from testing advanced learners (Solon et al., 2019) by including more complex items to testing specific grammatical target structures, e.g., *passives*, (Godfroid and Kim, 2021; Spada et al., 2015) to see if learners have acquired those structures.

To our knowledge, the first study using a corpus-based item bank in the context of EIT was carried out by Christensen et al. (2010). In this study, the authors used an annotated corpus for flexible item selection and then compared an existing EIT, using human-written items, with an EIT created with their automatic item selection tool. Interestingly, the automatically created version of their EIT had even higher correlation scores with an external measurement used in their study, the Spoken Language Achievement Test (SLAT), than the manually created EIT. While this is promising for the generation of items, the finding might also hint at different behaviour or validity characteristics of the two forms.

Using a corpus in the context of creating EIT items holds advantages: sentences from a corpus can be more authentic than items written for EIT use and, additionally, choosing sentences from a corpus is also less work than hand-crafting EIT items. However, this approach can also introduce a substantial amount of noise and effort: sentences can be out of context or inappropriate. Furthermore, items might be too long or overly complicated and require manual adaptation. A corpus might also be limited to a specific domain only,

which might be unfamiliar to some population of learners. Another difficulty is having control over potential target structures in the corpus sentences, which requires additional annotation of the corpus. A corpus-based item bank in the context of EIT therefore also places demands on test creators and does not entirely simplify the item creation process.

In summary, previous approaches in EIT relied on manual creation or corpus-based retrieval of items. Corpus-based methods are a step towards automation, however, they only offer a partial solution to the problems of scalability, effort, structural control and standardisation. These limitations encourage the exploration of alternative approaches, such as Automatic Item Generation (AIG).

2.3 Automatic Item Generation

AIG refers to the creation of (test) items using automated approaches such as machine learning and carries several advantages, among them more time- and resource-efficient item creation (Circi et al., 2023).

This approach finds application in several fields and contexts, for example, the creation of multiple choice items in a medical context (Gierl et al., 2012) or for generating reading comprehension questions (Huang and He, 2016; Sayin and Gierl, 2024). For generating items, cognitive models are employed, which model the domain knowledge, content, or skills needed to develop items (Gierl et al., 2012). Recent advancements in Large Language Models (LLMs) have created new approaches in AIG. While AIG usually involves the complex task of incorporating cognitive models in item creation, LLMs have enabled a less technically complex approach, involving few- or zero-shot prompting (Attali et al., 2022) and allowing for more flexibility and fewer constraints in the item creation process, for example by allowing the creation of a variety of item types in various subject domains (Tan et al., 2024). Generally, using prompt engineering with pre-trained models is a growing trend in the field (Song et al., 2025) and is additionally a feasible method for automatic item creation for people without a computational background, for example teachers and educators.

Beyond the generation of items, an equally important aspect in AIG is the evaluation and validation of test items. In the context of AIG with LLMs, item and measurement properties are often not reported or evaluated (Tan et al., 2024). Especially in the context of high-stakes tests, validation

and testing of items is crucial to make sure that the test actually tests what it is supposed to test. This is also important in the case of the EIT, as generated items may differ in linguistic structures or lengths when compared to established test items, potentially altering cognitive load for the participant and item difficulty.

Tests are often evaluated in terms of validity. Broadly speaking, validity can be defined as the "meaning of test scores" (Messick, 1995, p. 5), referring to the overall concept as describing whether a test measures what it intends to measure. Validity can be measured in several ways. An example is construct validity, an aspect of validity that tests if a test indeed measures the underlying construct it is trying to measure, or criterion validity, that measures how related test results are to another test (El-Hamamsy et al., 2022).

For the evaluation of AIG items, recent work has focused on automated approaches, but using a human-in-the-loop evaluation of items, for example, on the rating of the quality of generated items (Kim et al., 2025; Ma et al., 2025). Another possibility for assessing the suitability of items are psychometric methods such as Item Response Theory (IRT), which can inform about validity and quality in the context of AIG (Falcão et al., 2023).

Using AIG in the context of Elicited Imitation Tests certainly holds potential: it enables parallel testing, the generation of a large item bank, and supports the labour-intensive process of crafting items. While an earlier approach (Christensen et al., 2010) used corpora and annotation to find suitable items, LLMs can simplify this process by responding flexibly to certain demands, e.g. the presence of specific constructs or topics. This allows for more control than a corpus-based setup.

3 Present Study

Taken together, AIG seems to be promising in the context of EITs, allowing for flexibility and opening potential for the creation of parallel forms, which in turn can help with the process of standardisation. However, automatically generated items would need to be validated in order to make claims about their potential to be used in language assessment. To date, no study has systematically investigated whether LLM-generated items can be used as valid EIT items. To research this, we use an LLM-generated EIT, employing GPT-4o. We opted for GPT as it is the most widely adopted tool in ed-

ucation (Bhullar et al., 2024), accessible through a web interface and frequently used in AIG (Tan et al., 2024; Chan et al., 2025), which helps with comparability of the findings. This way, our findings are directly relevant to educators and students in practice. For comparability, we will concentrate on parallel forms in particular. Since grammatical structures and complexity are of specific interest to EIT items, we will investigate item properties and measurement properties in two ways: evaluating items and tests in terms of their linguistic and psychometric properties. The research questions are as follows:

1. To what extent do the properties of GPT-generated test items differ from those of established test items? More specifically,
 - (a) Are they different in terms of linguistic complexity?
 - (b) Are they different in terms of grammatical structures used?
2. How do the automatically generated tests compare to established tests in terms of validity?

4 Method

4.1 Materials

Three EITs were used in this study: (1) An established EIT by Ortega et al. (2002), (2) the parallel version by Wu et al. (2022) and (3) an EIT generated by GPT-4o. The items of the GPT-generated EIT can be found in Appendix A.

Our automatic item generation approach was as follows: we prompted GPT-4o to create an EIT with 30 items, varying in length, starting at 7 syllables. We employed few-shot prompting for automatic item generation, including the items of the original EIT by Ortega et al. (2002) as well as their syllable count as examples. The model output, 30 items, was used for the resulting GPT-generated EIT. The prompt can be found in Appendix C.

The tests were all uploaded to ILAP (Löber et al., 2024), an assessment platform, where participants could take the tests on their own devices via a web browser. All EIT tests used the same procedure: The participants saw instructions about the test, followed by three practice items that were not counted towards the final score. Then, a notification about the start of the test followed. The test then began. After each sentence, there was a beep followed by a four-second pause. Participants then had eight seconds to repeat the sentence they had just heard.

Audio stimuli for all items on the three EITs were created with the Amazon AWS text-to-speech service, which had been integrated into the platform.

Furthermore, we administered questionnaires about participants' demographic background and a C-Test as a measure for general English language proficiency taken from Ishihara et al. (2003). The C-Test was programmed with jsPsych (De Leeuw, 2015) in JavaScript and integrated into the platform used for EIT data collection. The test consisted of three text passages with half of every fifth or sixth word removed. The maximum score on the C-test was 50 points (15 points for each of the first 2 passages and 20 points for the last passage).

The EITs were scored automatically, using the ASR and string matching approach by Isbell et al. (2023). We used accuracy based on Mean Error Rate (MER) - $(100 - \text{MER})$ - as the scoring metric (McGuire and Larson-Hall, 2025). We then mapped our continuous scores to ordinal scores for better comparability with the ordinal scoring scheme by Ortega et al. (2002), encompassing scores from 0-4. We followed the approach taken by Isbell et al. (2023) for mapping the scores, which can be found below in Table 1. For total scores, we took the sum of all item scores. The maximum score on the EITs was 120.

Ordinal score	Continuous score
4	91 - 100
3	71 - 90
2	41 - 70
1	21 - 40
0	20 or below

Table 1: The ordinal scoring scheme, mapping continuous scores to ordinal scores

The C-Test was also automatically scored, assigning binary scores (1 or 0) to each gap. This was done by employing string matching between the participants' response to the expected response for each gap. We then used the sum of all answers as the total score. The maximum score on the C-Test was 50.

4.2 Study Design

The study ran for four weeks. Each week, a teacher administered an EIT to a class in a randomised order, e.g. class 1 completing EIT 1 in week 1, EIT 2 in week 2 and EIT 3 in week 3 while class 2

completed EIT 2 in week 1, EIT 3 in week 2 and EIT 1 in week 3, etc.

Students created an individual accounts on the platform used for testing. The EITs were implemented into the platform by the researchers, where they could be administered to the students with an access code. Before being able to access any test, students were shown a consent form where they could indicate whether they consented or not to their data being used for study purposes. Furthermore, we also implemented the questionnaire and C-Test on the platform. These could be taken by students at any time during the course of the 4-week study.

4.3 Participants

Participants were 73 undergraduate students from a university in China, majoring in English. 60 of the students were female and the mean age was 20.1 years (SD = 1.55), ranging from 18 to 25 years of age. All students reported Chinese as their native language and the majority (72%) reported English as their second language. Students' self-reported English proficiency was intermediate to upper intermediate.

4.4 Analysis

Data were analysed with R (version 4.5.0), using the packages stringr (Wickham, 2019), lme4 (Bates et al., 2015), lavaan (Rosseel, 2012), mirt (Chalmers, 2012) and koRpus (Michalke et al., 2021). We also analysed all test items in terms of the grammatical structures and corresponding learner levels with the rule-based annotation tool POLKE (Sagirov and Chen, 2025). POLKE annotates text with grammatical structures and their corresponding Common European Framework of References (CEFR) (Council of Europe, 2001) learner levels (A1-C2) and is based on the English Grammar Profile (O'Keeffe and Mark, 2017). In addition to the grammatical structures and learner levels, the tool also supplies super-categories of structures (e.g. nouns, verbs, adjectives) as well as sub-categories (e.g. types, linking, phrasal).

In addition, we analysed the EIT items in terms of their linguistic complexity, using the tool CTAP (Chen and Meurers, 2016) as well as the stringr and koRpus packages in R. For measure selection, we followed Bulté et al. (2025)'s summary of core and noncore measures of complexity, selecting six measures. We also analysed word frequency, using the SUBTLEXus log-10 frequency of all tokens.

The full overview of complexity measures used can be found in Appendix B.

Furthermore, we employed Item Response Theory (IRT) and Confirmatory Factor Analysis (CFA) as well as correlations to analyse the test properties of the 3 EITs.

5 Results

5.1 Descriptive Results

Not all participants completed every measure. Specifically, 68 participants finished the background questionnaire, 62 the C-Test, 63 EIT 1, 61 EIT 2 and 62 EIT 3, with 53 participants completing all 3 EITs. We analysed all data available to us, which meant that for example, the correlation analyses were conducted with only participants who completed all measures included in the analysis.

The mean score on the C-Test was 38.96 (SD = 11.01). Mean scores on the EITs were similar: EIT 1 (59.6, SD = 34.50) and EIT 3 (57.82, SD = 31.20) displayed highly similar scores, whereas EIT 2 (51.58, SD = 35.78) showed a slightly lower mean score. EIT scores showed a high range, 11-113 for EIT 1, 5-112 for EIT 2 and 5-113 for EIT 3. The tests themselves displayed high levels of similarity: The mean syllable length was between 10.4 and 10.97 syllables for all three tests and the mean sentence length was between 14.93 and 15.2 for all three tests. Descriptive statistics for the scores on the three tests, including standard deviations (SD), can be found in Table 2 and 3 below.

Test	n	Mean score	SD
Ortega (2002)	63	59.58	34.50
Wu et al. (2022)	61	51.58	35.78
GPT-4o	62	57.82	31.20

Table 2: Number of participants, mean score and standard deviation (SD) for the three tests.

Test	Mean SL	SD	Mean SC	SD
Ortega (2002)	10.83	2.59	14.93	3.34
Wu et al. (2022)	10.97	2.62	15.03	3.78
GPT-4o	10.40	2.62	15.20	3.89

Table 3: Descriptive statistics for test properties. SL = sentence length, SC = syllable count.

5.2 Items

We first analysed the POLKE annotations in terms of the distribution of super-categories. We excluded

categories with counts < 5 in order to meet the assumptions of chi-square tests. These were the categories future, passives, questions and reported speech. The distribution of the remaining categories differed significantly between tests, $\chi^2(22) = 39.839$, $p = 0.011$. Following this, we inspected the residuals of the chi-square test. The GPT-generated tests produced fewer structures belonging to the adverbs and present category and more structures belonging to the preposition category than EIT 1 and EIT 2. Figure 1 shows the distribution of super-categories.

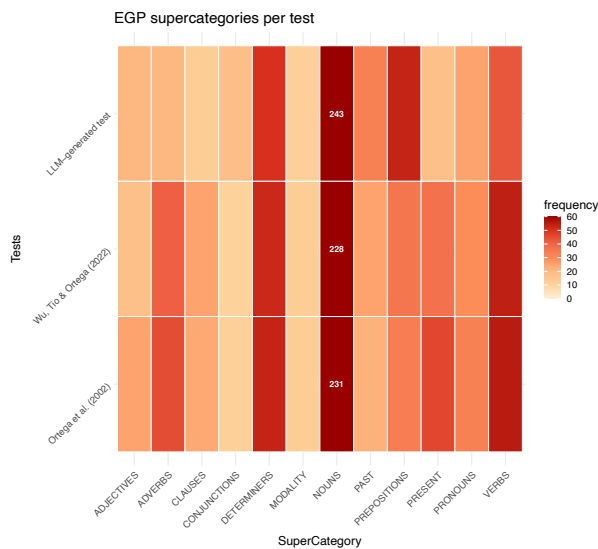


Figure 1: The distribution of super-categories for all three tests.

After that, we analysed whether there were differences in the distribution of CEFR learner levels within the POLKE-annotated grammar structures between the tests. The distribution of learner levels did not significantly differ between the tests, $\chi^2(10) = 9.8922$, $p = 0.45$. Figure 2 shows this distribution.

We then divided the complexity measures into sub-measures of syntactic (Table 4) and lexical complexity (Table 5). The results for each can be found in the tables below.

Prior to statistical testing, we used Shapiro-Wilk Tests to check for the distribution of data. If the tests were significant, we rejected the assumption of normality and used non-parametric tests. We first ran Kruskal-Wallis tests on the syntactic complexity measures. Three measures did not differ significantly between conditions: mean words per phrase ($p = 0.09$), mean phrases per clause ($p = 0.70$) and mean clauses per T-Unit ($p = 0.09$). One

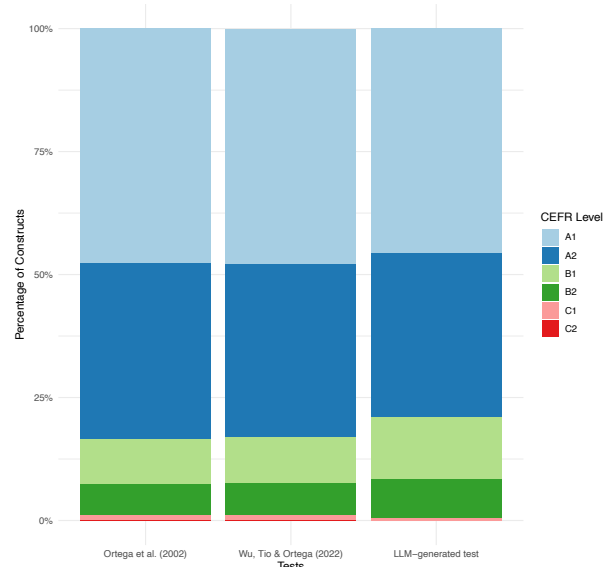


Figure 2: The distribution of learner levels associated with grammatical structures for all three tests, in percentages.

difference was significant: the normalised rate of occurrence of dependent clauses differed significantly between the tests, $H(2) = 6.09$, $p = 0.048$. We used Dunn's post hoc test with Bonferroni corrections to assess which groups differed. However, after adjusting, there was no significant difference between the Ortega et al. (2002) test and the GPT-generated form ($p = 0.177$) or between the Wu et al. (2022) form and the GPT-generated form ($p = 0.062$).

For lexical complexity, a Kruskal-Wallis test showed that the mean word length did not vary significantly between tests, $H(2) = 2.93$, $p = 0.231$. Word frequency also did not differ significantly between tests ($p = 0.178$).

Test	MW phrase	MP clause
Ortega (2002)	1.47	4.17
Wu et al. (2022)	1.59	3.98
GPT-4o	1.68	4.20
Test	MC T-Unit	N. ROC
Ortega (2002)	1.73	0.44
Wu et al. (2022)	1.76	0.47
GPT-4o	1.42	0.27

Table 4: Syntactic complexity mean scores for all three tests. MW = mean words, MP = mean phrases, MC = mean clauses, N. ROC = Normalised rate of occurrence of dependent clauses.

Test	Mean WL	MATTR	WF
Ortega (2002)	3.95	0.92	4.71
Wu et al. (2022)	4.01	0.91	4.60
GPT-4o	4.35	0.92	4.55

Table 5: Lexical complexity mean scores for the three tests. MATTR = moving-average type-token ratio, WL = word length, WF = word frequency.

5.3 Validity

All three tests displayed very high internal consistency, $\alpha = 0.98$.

We first calculated Spearman correlations of the scores between tests. EIT 1 and 3 showed a correlation coefficient of $\rho = 0.80$ (shared variance: 0.64), EIT 1 and 2 showed a correlation coefficient of $\rho = 0.76$ (shared variance: 0.58), and EIT 2 and 3 correlated with a correlation coefficient of $\rho = 0.76$ (shared variance: 0.58).

Using IRT, we then fitted a 1PL graded response model to the 3 EITs and used common-person linking to obtain discrimination values for the items. All items showed acceptable discrimination values higher than 0.39 (Popham, 2000). We compared the discrimination values between tests, which differed significantly, $H(2) = 6.61$, $p = 0.037$. Dunn’s post hoc test with Bonferroni corrections showed that the discrimination values differed only between the Wu et al. (2022) and GPT-generated form, $p = 0.038$, with the GPT-generated form showing higher discrimination values, but not between the Ortega et al. (2002) and GPT-generated form ($p = 0.22$) or the Ortega et al. (2002) and Wu et al. (2022) form ($p = 1$).

Focusing on the test level next, we employed confirmatory factor analysis (CFA) to obtain standardised factor loadings for the tests. We created a one-factor CFA model, using Full Information Maximum Likelihood (FIML) for missing data, with all three tests loading on the same latent variable, namely proficiency. The model was just-identified ($df = 0$), so we could not assess model fit. We then compared the standardised factor loadings of the tests. All tests displayed strong and significant factor loadings, indicating that they are connected in a highly similar way to the construct proficiency. The factor loadings can be found in table 6.

To obtain a measure of criterion validity, we correlated participants’ final scores on the EIT with their final scores on the C-Test using Spearman correlation. All EITs showed a weak to moderate

Test	Factor loadings
Ortega (2002)	0.89
Wu et al. (2022)	0.85
GPT-4o	0.93

Table 6: Standardised factor loadings per test.

correlation with the C-Test, $\rho = 0.26$ for EIT 1, $\rho = 0.35$ for EIT 2, and $\rho = 0.29$ for EIT 3. Using Fisher’s r-to-z transformation, we tested if the correlations differed between tests. There was no significant difference between EIT 1 and EIT 2 ($p = 0.55$), EIT 2 and 3 ($p = 0.72$) or EIT 1 and 3 ($p = 0.81$).

5.4 Scores

To find out if participants’ final test scores differed significantly between tests, we ran a mixed linear model, modeling participants as random effects to account for individual differences between users. The resulting linear mixed model with users as random effects showed a significant main effect of test between EIT 1 and EIT 2 scores ($p = 0.021$), and no difference between EIT 1 and EIT 3 ($p = 0.66$). Participants therefore had a significantly lower score on EIT 2 than on EIT 1.

6 Discussion and Conclusion

In this study, we explored the potential for automatic item generation for parallel forms with GPT-generated items in the context of Elicited Imitation. Our results are promising: the GPT-generated test behaved very similarly to the original Ortega et al. (2002) form as well as the parallel form (Wu et al., 2022), displaying highly similar linguistic properties, high internal reliability as well as similar factor loadings and construct validity.

There were slight differences between the tests (human-written vs GPT-generated) when it came to the grammatical constructs used: The GPT-generated test displayed a lower number of adverbs and present (tense) and higher numbers of prepositions than the other two tests. However, there was no difference between the three tests in the learner levels associated with the grammatical constructs, potentially an indicator for the non-significant difference in the difficulty of items between the tests. In other words, the differences found in the use of grammatical structures did not seem to contribute to the difficulty of the tests. We should also add that we did not specifically prompt for following

the grammatical constructs used closely. Interestingly, the different grammatical structures do not seem to influence item or test difficulty.

The complexity analysis showed that all three tests score very similarly on all measures. The only significantly different measure, normalised rate of occurrence of dependent clauses, turned out to be non-significant in the post hoc tests.

In summary, the GPT-generated test deviates from the original EIT and the human-written parallel form on a few grammatical structures and not on complexity measures, but the deviations on grammatical structures do not seem to add to overall test difficulty of the GPT-generated form or interfere with form equivalence. This finding should be further investigated in future studies, focusing on item and test difficulty on EITs. While some studies have already investigated this topic (Hendrickson et al., 2010; Campfield, 2017) and found effects depending on the sentence length or lexical complexity, future studies might want to explore that topic further.

A slightly different picture emerged when looking at our psychometric analysis. While the Ortega et al. (2002) and GPT-generated form seem highly similar, the Wu et al. (2022) form displayed some deviations from both forms: the IRT analysis showed lower item discrimination scores than for the GPT-generated form. In terms of scores, the parallel form from Wu et al. (2022) showed a lower mean score. This is in line with the findings from the parallel forms study (Wu et al., 2022), where the authors found an 8% score difference between the parallel form and the Ortega et al. (2002) form. Between those two tests, we found a score difference of 14%. Interestingly, the TTS voice used in our tests was the same across tests, pointing to no added difficulty introduced by speaker variability as Wu et al. (2022) suspected as one of the culprits for enhanced test difficulty. The lower score on this test must therefore be for reasons beyond speaker variability and the complexity measures employed in this study. We also reported less shared variance between the tests than in Wu et al. (2022). This finding might be due to the different population used in this study.

An unexpected finding also emerged from our data: the correlation of the C-Test with performance on the EIT was quite low (weak correlation of $\rho = 0.26$ and $\rho = 0.29$ on EIT 1 and EIT 3, respectively, and moderate ($\rho = 0.35$) on EIT 2), which is contrary to what other studies found

($r = 0.69$ in Davis and Norris, 2021, $r = 0.50$ in Spada et al., 2015). A C-Test counts as a measure of holistic proficiency, but due to the nature of the C-Test, which is more geared towards writing and reading, it can be argued that the sub-skills of listening and speaking are not included, which is what the EIT taps into. These two sub-skills might have been developed less strongly when compared to writing and reading for our participants, which might explain the weak correlation. In our case, the mean scores on the C-Test and the EITs also point to C-Tests being easier for our participants than the EIT.

Taken together, our findings show that an LLM-generated test holds potential in creating valid EIT items in a parallel forms context. This finding could enable the scaling of tests, making it possible to create more parallel forms with LLM-generated items. Furthermore, this can potentially help to tackle the problem of standardisation in language research (Isbell and Son, 2022), by enabling the creation – at least in the case of EIT – of comparable forms to existing EITs. For future studies, it would be interesting to see if our findings would be applicable across language contexts. While we only concentrated on English in this study, performance of LLMs might be less consistent beyond that and it is unclear if the performance of GPT can be carried over to other languages. Since all the available parallel and comparable EIT forms across languages are created manually (e.g. Wu and Ortega, 2013; Bowden, 2016), studies could shed more light on the performance of automatically generated items across languages.

7 Limitations

A few limitations of this research should be mentioned. First of all, the number of participants ($n = 73$) was quite low, leading to less generalisable results. Due to this low number of participants, we were not able to carry out more sophisticated analyses, e.g. Differential Item Functioning and our IRT analysis should be taken as preliminary. Having more participants at hand, and possibly a more heterogeneous group of participants, would undoubtedly shed more light on validity and parallel form equivalence.

Furthermore, due to the nature of our online experiment, we experienced a number of dropout cases and missing data from the background questionnaire.

As already pointed out in the previous section, we also did not compare performance across different LLMs. GPT is one of the most readily available options and widely used in the context of education (Doughty et al., 2024; Vanzo et al., 2025) and is also relevant to educators, teachers and test creators. However, the performance of other models would also be of interest for this strand of research.

We also only tested 30 LLM-generated sentences. While our results seem promising for parallel test creation, future research should test a larger number of items to evaluate if our findings are generalisable to different contexts, e.g. repeated or adaptive testing.

Lastly, we did not take the ability of models to produce certain grammatical constructs on demand into account. While we found that the GPT-generated test mostly showed strong similarities to the two human-written forms in terms of super-categories, we didn't explicitly prompt for it. Further research could concentrate on this ability of LLMs, as this is central to some EITs targeting specific grammatical constructs and can shed light on large language models' abilities to produce grammatical constructs.

Acknowledgements

This work was supported by the German Ministry of Education and Science (BMBF) under Grant number 01IS22076. We would like to thank the anonymous reviewers for their helpful comments.

References

- Yigal Attali, Andrew Runge, Geoffrey T LaFlair, Kevin Yancey, Sarah Goodwin, Yena Park, and Alina A Von Davier. 2022. [The interactive reading task: Transformer-based automatic item generation](#). *Frontiers in Artificial Intelligence*, 5:903077.
- Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, Bin Dai, Gabor Grothendieck, Peter Green, and M Ben Bolker. 2015. Package 'lme4'. *convergence*, 12(1):2.
- Pritpal Singh Bhullar, Mahesh Joshi, and Ritesh Chugh. 2024. [Chatgpt in higher education-a synthesis of the literature and a future research agenda](#). *Education and Information Technologies*, 29(16):21501–21522.
- Harriet Wood Bowden. 2016. [Assessing second-language oral proficiency for research: The Spanish elicited imitation task](#). *Studies in Second Language Acquisition*, 38(4):647–675.
- Bram Bulté, Alex Housen, and Gabriele Pallotti. 2025. [Complexity and difficulty in second language acquisition: A theoretical and methodological overview](#). *Language Learning*, 75(2):533–574.
- Dorota E. Campfield. 2017. [Lexical difficulty – using elicited imitation to study child L2](#). *Language Testing*, 34(2):197–221.
- R Philip Chalmers. 2012. mirt: A multidimensional item response theory package for the r environment. *Journal of statistical Software*, 48:1–29.
- Kuang Wen Chan, Farhan Ali, Joonhyeong Park, Kah Shen Brandon Sham, Erdalyn Yeh Thong Tan, Francis Woon Chien Chong, Kun Qian, and Guan Kheng Sze. 2025. Automatic item generation in various stem subjects using large language model prompting. *Computers and Education: Artificial Intelligence*, 8:100344.
- 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 at COLING*. The International Committee on Computational Linguistics.
- Mihail Chifligarov, Jammila Laâguidi, Max Schellenberg, Alexander Dill, Anna Timukova, Anastasia Drackert, and Ronja Laarmann-Quante. 2025. [Automated scoring of a German written elicited imitation test](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 237–247, Vienna, Austria. Association for Computational Linguistics.
- Carl Christensen, Ross Hendrickson, and Deryle Lonsdale. 2010. Principled construction of elicited imitation tests. In *Proceedings of the 7th conference on international language resources and evaluation (LREC)*.
- Ruhan Circi, Juanita Hicks, and Emmanuel Sikali. 2023. [Automatic item generation: foundations and machine learning-based approaches for assessments](#). In *Frontiers in Education*, volume 8, page 858273.
- Council of Europe. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge University Press.
- Larry Davis and John Norris. 2021. [Developing an innovative elicited imitation task for efficient english proficiency assessment](#). *ETS Research Report Series*, 2021(1):1–30.
- Joshua R De Leeuw. 2015. [jspsych: A Javascript library for creating behavioral experiments in a Web browser](#). *Behavior Research Methods*, 47(1):1–12.
- Jacob Doughty, Zipiao Wan, Anishka Bompelli, Jubahed Qayum, Taozhi Wang, Juran Zhang, Yujia Zheng, Aidan Doyle, Pragnya Sridhar, Arav Agarwal, and 1 others. 2024. [A comparative study of AI-generated \(GPT-4\) and human-crafted MCQs in programming](#).

- education. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 114–123.
- Laila El-Hamamsy, María Zapata-Cáceres, Estefanía Martín Barroso, Francesco Mondada, Jessica Dehler Zufferey, and Barbara Bruno. 2022. The competent computational thinking test: Development and validation of an unplugged computational thinking test for upper primary school. *Journal of Educational Computing Research*, 60(7):1818–1866.
- Rosemary Erlam. 2006. Elicited imitation as a measure of L2 implicit knowledge: An empirical validation study. *Applied Linguistics*, 27(3):464–491.
- Filipe Falcão, Daniela Marques Pereira, Nuno Gonçalves, Andre De Champlain, Patrício Costa, and José Miguel Pêgo. 2023. A suggestive approach for assessing item quality, usability and validity of Automatic Item Generation. *Advances in Health Sciences Education*, 28(5):1441–1465.
- Mark J Gierl, Hollis Lai, and Simon R Turner. 2012. Using automatic item generation to create multiple-choice test items. *Medical Education*, 46(8):757–765.
- Aline Godfroid and Kathy Minhye Kim. 2021. The contribution of implicit-statistical learning aptitude to implicit second-language knowledge. *Studies in Second Language Acquisition*, 43(3):606–634.
- Ross Hendrickson, Meghan Aitken, Jeremiah McGhee, and Aaron Johnson. 2010. What makes an item difficult? A syntactic, lexical, and morphological study of elicited imitation test items. In *Selected Proceedings of the 2008 Second Language Research Forum*, pages 48–56. Cascadilla Proceedings Project Somerville, MA.
- Yan Huang and Lianzhen He. 2016. Automatic generation of short answer questions for reading comprehension assessment. *Natural Language Engineering*, 22(3):457–489.
- Daniel R. Isbell, Kathy Minhye Kim, and Xiaobin Chen. 2023. Exploring the potential of automated speech recognition for scoring the Korean Elicited Imitation Test. *Research Methods in Applied Linguistics*, 2(3):100076.
- Daniel R. Isbell and Young-A Son. 2022. Measurement properties of a standardized elicited imitation test: An integrative data analysis. *Studies in Second Language Acquisition*, 44(3):859–885.
- Kenji Ishihara, Elizabeth Hiser, and Tae Okada. 2003. Modifying C-test for practical purposes. *Doshisha Studies in Language and Culture*, 5(4):539–568.
- Euigyum Kim, Seewoo Li, Salah Khalil, and Hyo Jeong Shin. 2025. STAIR-AIG: Optimizing the automated item generation process through human-AI collaboration for critical thinking assessment. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 920–930, Vienna, Austria. Association for Computational Linguistics.
- Kathy Minhye Kim, Xiaoyi Liu, Daniel R Isbell, and Xiaobin Chen. 2024. A comparison of lab-and web-based elicited imitation: Insights from explicit-implicit L2 grammar knowledge and L2 proficiency. *Studies in Second Language Acquisition*, pages 1–22.
- Marianne Klem, Monica Melby-Lervåg, Bente Hagtvet, Solveig-Alma Halaas Lyster, Jan-Eric Gustafsson, and Charles Hulme. 2015. Sentence repetition is a measure of children’s language skills rather than working memory limitations. *Developmental Science*, 18(1):146–154.
- Sarah Löber, Björn Rudzewitz, Daniela Verratti Souto, Luisa Ribeiro-Flucht, and Xiaobin Chen. 2024. Developing a Web-Based Intelligent Language Assessment Platform powered by Natural Language Processing Technologies. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 126–136.
- Wanjing (Anyu) Ma, Michael Flor, and Zuowei Wang. 2025. Automatic generation of inference making questions for reading comprehension assessments. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 398–414, Vienna, Austria. Association for Computational Linguistics.
- Michael McGuire and Jenifer Larson-Hall. 2025. Assessing Whisper automatic speech recognition and WER scoring for elicited imitation: Steps toward automation. *Research Methods in Applied Linguistics*, 4(1):100197.
- Samuel Messick. 1995. Validity of psychological assessment: Validation of inferences from persons’ responses and performances as scientific inquiry into score meaning. *American Psychologist*, 50(9):741.
- Meik Michalke, Earl Brown, Alberto Mirisola, Alexandre Brulet, Laura Hauser, and Maintainer Meik Michalke. 2021. Package ‘korpus’.
- Lourdes Ortega, Noriko Iwashita, John M Norris, and Sara Rabie. 2002. An investigation of elicited imitation tasks in crosslinguistic SLA research. In *Second Language Research Forum, Toronto*, pages 3–6. Paper presentation.
- 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.
- W James Popham. 2000. The mismeasurement of educational quality. *School Administrator*, 57(11):12–15.
- Yves Rosseel. 2012. lavaan: An r package for structural equation modeling. *Journal of statistical software*, 48:1–36.

- Nelly Sagirov and Xiaobin Chen. 2025. POLKE: A system for comprehensively annotating pedagogically-oriented grammatical structure use in language production. *Manuscript submitted for publication to Behavior Research Methods*.
- Ayfer Sayin and Mark Gierl. 2024. [Using OpenAI GPT to generate reading comprehension items](#). *Educational Measurement: Issues and Practice*, 43(1):5–18.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine Learning–Driven Language Assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.
- Megan Solon, Hae In Park, Carly Henderson, and Marzieh Dehghan-Chaleshtori. 2019. [Revisiting the Spanish Elicited Imitation Task: A tool for assessing advanced language learners?](#) *Studies in Second Language Acquisition*, 41(5):1027–1053.
- Yishen Song, Junlei Du, and Qinhua Zheng. 2025. [Automatic item generation for educational assessments: a systematic literature review](#). *Interactive Learning Environments*, pages 1–20.
- Nina Spada, Julie Li-Ju Shiu, and Yasuyo Tomita. 2015. [Validating an elicited imitation task as a measure of implicit knowledge: Comparisons with other validation studies](#). *Language Learning*, 65(3):723–751.
- Bin Tan, Nour Armoush, Elisabetta Mazzullo, Okan Bulut, and Mark Gierl. 2024. [A review of automatic item generation techniques leveraging large language models](#). *International Journal of Assessment Tools in Education*, 12(2):317–340.
- Alessandro Vanzo, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2025. [GPT-4 as a homework tutor can improve student engagement and learning outcomes](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31119–31136.
- Hadley Wickham. 2019. Package ‘stringr’. <http://stringr.tidyverse.org>.
- Shu-Ling Wu and Lourdes Ortega. 2013. [Measuring global oral proficiency in SLA research: A new elicited imitation test of L2 Chinese](#). *Foreign Language Annals*, 46(4):680–704.
- Shu-Ling Wu, Yee Pin Tio, and Lourdes Ortega. 2022. [Elicited imitation as a measure of L2 proficiency: New insights from a comparison of two L2 english parallel forms](#). *Studies in Second Language Acquisition*, 44(1):271–300.
- Xun Yan, Yuyun Lei, and Chilin Shih. 2020. [A corpus-driven, curriculum-based chinese elicited imitation test in US universities](#). *Foreign Language Annals*, 53(4):704–732.

A Appendix A

GPT-generated EIT test items

1. She is buying a new jacket.
2. The blue car is parked outside.
3. The store closes early on Sundays.
4. He usually takes the bus to work.
5. I don’t know why she left so suddenly.
6. We had a great time at the beach yesterday.
7. The restaurant downtown serves excellent seafood.
8. She was excited to finally start her new job.
9. The dog that lives next door barks all the time.
10. I hope to visit my grandmother this weekend.
11. The weather in this city changes very quickly.
12. She spent the whole afternoon reading a good book.
13. My neighbor’s cat always sits by the window and watches birds.
14. The teacher gave us an interesting assignment about history.
15. The bakery on the corner sells delicious chocolate cake.
16. I wish I could have stayed longer at the party last night.
17. He told me that he would help me move to my new apartment.
18. The little girl with curly hair is playing in the garden.
19. We decided to take a road trip through the mountains next week.
20. If I had known about the meeting, I would have arrived earlier.
21. She was surprised to see her childhood friend after so many years.
22. The scientist explained how the new discovery could change medicine.

23. He was looking for his wallet but couldn't remember where he left it.
24. The old library downtown has a huge collection of rare books.
25. I didn't expect the movie to be so emotional and thought-provoking.
26. The woman who just moved in next door seems very friendly and kind.
27. We were planning to go hiking, but the weather turned out terrible.
28. The students who studied all night found the exam easier than expected.
29. If I had known how difficult this course would be, I might have chosen another.
30. The train that was supposed to arrive at noon has been delayed due to bad weather.

B Appendix B

Complexity measures employed in this study

- Mean word length
- MATTR of lemmas
- Mean words per phrase
- Mean phrases per clause
- Mean clauses per T-Unit
- Normalised rate of occurrence of dependent clauses
- SUBTLEXus log-10 mean frequency of all tokens

C Appendix C

Could you create an Elicited Imitation Test? Please create 30 items, varying in length, start with the shortest one, then gradually generate longer items. Start at 7 syllables.

Below, you will find an example of an existing EIT (in brackets are the amount of syllables of the sentences).

Please only respond with your generated items.
{examples}

Comparative Evaluation of AI-Generated vs. Expert-written Answer Explanations for a Medical Education Self-Assessment

Yiyun Zhou, Francis O'Donnell, and Victoria Yaneva
National Board of Medical Examiners, Philadelphia, USA
{YYZhou, FODonnell, VYaneva}@nbme.org

Abstract

Answer explanations for medical multiple-choice questions (MCQs) are a valuable learning tool, but producing them is resource-intensive. Writing high-quality explanations requires specialized medical expertise and careful alignment with the keyed answer, distractors, and the clinical vignette. This paper evaluates whether a template-aware, retrieval-guided large language model (LLM) workflow can support this production task in a real formative assessment setting. Using a 50-item medical education self-assessment, we compared AI-generated and expert-written MCQ explanations in a blinded study involving eight medical faculty and sixteen medical students. Each participant rated 25 of 50 paired explanations on clarity, amount of information, and structure. The clearest empirical difference was in amount of information: AI-generated explanations were rated significantly higher than expert-written explanations in a cumulative link mixed model analysis (OR = 1.99, 95% CI [1.33, 2.99], $p = 0.001$). Ratings of clarity and structure did not differ significantly between conditions. Based on faculty ratings, a smaller proportion of AI-generated explanations were judged to require correction (20%) compared with expert-written explanations (38%). These findings suggest that AI can reduce first-draft authoring effort in explanation writing while still requiring expert review to ensure content accuracy.

1 Introduction

Multiple-choice questions (MCQs) are a central component of clinical education and assessment, yet their instructional potential often extends beyond the selection of a correct answer. In many clinical learning tools, MCQs are accompanied by narrative explanations that justify why the correct option is appropriate and why the alternative options are incorrect (see Appendix A for an example

MCQ and explanation). These explanations typically outline the relevant clinical concepts, diagnostic reasoning, or guideline-based considerations that underlie the item, making the expert reasoning process more transparent to learners. By explicitly articulating the logic behind answer choices, answer explanations can help students connect factual knowledge with clinical decision-making, reinforce key principles, and clarify common misconceptions. As a result, they have the potential to transform MCQs from purely evaluative instruments into powerful learning tools that support the development of clinical reasoning.

While valuable as learning support instruments, writing answer explanations for medical self-assessment questions is resource-intensive, which limits the availability of such materials. This is largely due to the strict criteria these explanations must satisfy to be pedagogically useful. Each explanation must justify the correct answer, explain why distractors are incorrect, remain faithful to the clinical vignette, and conform to an editorial style that supports instruction (Ch'en et al., 2025). In practice, therefore, producing high-quality explanations involves more than clinical expertise alone; it also requires careful editing and review to ensure clarity, accuracy, and consistency.

Large language models (LLMs) offer a possible way to reduce this drafting burden. Prior work has shown strong LLM performance on medical question answering and medical explanation tasks (Singhal et al., 2023, 2025). However, most of this literature has focused on answer accuracy and explanation correctness on benchmark datasets rather than the production of pedagogically valuable explanation drafts in real assessment workflows (Kim et al., 2024; Alonso et al., 2024; Chen et al., 2025). That gap matters because educational explanation writing requires not only domain reasoning, but also adherence to instructional goals, editorial conventions, formatting requirements, and expert qual-

ity control; it is also a workflow task shaped by editorial and quality control procedures.

This paper makes two main contributions. First, we develop a pipeline for generating answer explanations for clinical MCQs using AI. The pipeline is designed to produce explanations that address both the correct answer and the distractors while adhering to instructional conventions used in medical education materials. Second, we evaluate the quality of the generated explanations in a comparative study with expert-written explanations. Using blinded evaluations from 8 clinical faculty members and 16 medical students, we find that AI-generated explanations perform comparably to or better than expert-written explanations on clarity, amount of information, and structure.

2 Related Work

Prior work shows that LLMs can perform strongly on medical question answering (QA) benchmarks and that domain adaptation and prompting can further improve performance on medical QA and long-form medical responses, but this literature has largely emphasized answer accuracy rather than the quality, reliability, or educational usefulness of the accompanying explanations (Jin et al., 2021; Singhal et al., 2023, 2025; Chen et al., 2025). This motivates our focus on evaluating explanation quality directly rather than treating the explanations as a by-product of answer prediction.

Our method is also related to work on retrieval-augmented generation, in-context example retrieval, and post-generation refinement and verification. Retrieval can improve knowledge-intensive generation, and the choice of in-context examples can substantially affect downstream LLM performance (Lewis et al., 2020; Wang et al., 2024). Our expert-guided prompt iteration and post-generation checks are also informed by work on automatic prompt optimization, self-refinement, and verification-based decoding (Pryzant et al., 2023; Madaan et al., 2023; Dhuliawala et al., 2024). However, prior work does not directly address the question studied here: whether a retrieval- and prompt-based LLM workflow can generate answer explanations for real medical self-assessment items that are acceptable under expert review and comparable to expert-written explanations.

3 Dataset

The study used two data sets, each consisting of 50 MCQs, from a self-assessment program for medical students in the clinical stage of their training.

Set 1 served as the reference set and included MCQs with corresponding explanations that were used in the system prompts. Set 2 was the experimental set. Although it also contained expert-written explanations, these were not used during AI generation; instead, they were reserved for the later human evaluation, where AI-generated explanations were compared with the expert-written explanations for Set 2.

In addition to the two sets of items and explanations, two internally developed explanation templates reflecting existing editorial standards were provided – a *General* template for areas like basic science and biostatistics, and a *Clinical* template for areas that focus on patient management workflows. These templates guided the expected organization and level of detail in the generated explanations.

Each item within the sets included three distinct metadata tags: *Content Area* - the specific medical topic, coded hierarchically, e.g., Multi: acid-base disorders; *Competency Group* - the targeted physician competency, e.g., B2 Laboratory/diagnostic studies; and *Template Type* - the explanation template structure as either General or Clinical, determined by an LLM. There were 49 content areas and 13 competency groups represented in the dataset.

4 Methodology

When creating answer explanations for MCQs, we face three main challenges.

First, different explanations need to follow different templates depending on their content, as explained in the previous section. Therefore, one of our first tasks given a new item is to perform a binary classification step to correctly classify it into the template most appropriate for its content. Our approach to this task is described in section 4.1. Next, we need to determine the most appropriate items from our reference set that can be used for few-shot prompting. This task is challenging, because the items in the reference set cover multiple topics, content areas and tasks, and relevance can be determined across multiple dimensions. Our approach to this task is described in section 4.2. Finally, the generation is performed using an agentic

structure with *Expert-in-the-Loop* (EITL) calibration, as described in Section 4.3.

Figure 1 provides an overview of the approach and the following sections describe each step.

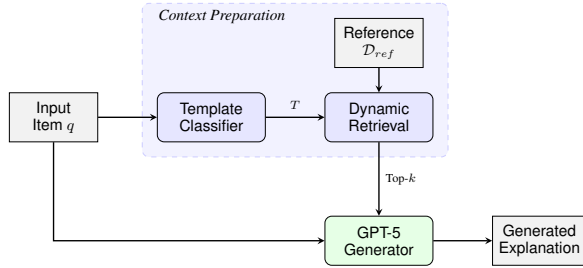


Figure 1: The workflow Architecture. Input items are routed via a classifier to retrieve context, which is then fed alongside the original input into the generator.

4.1 Dataset Stratification and Template Classification

To optimize the generation process for domain specificity, we employ a schema-aware preprocessing strategy that segments the reference corpus (Set 1), $\mathcal{D}_{ref}$, into distinct reasoning templates - *Clinical* template and *General* template. Correctly routing input items to these templates is critical for correctly organizing the information in the explanations. We implement a hybrid classification pipeline that processes each target item q to assign a template category $T \in \{T_{clinical}, T_{general}\}$. The primary classification is performed by a large language model (GPT-5) which analyzes the semantic content and the metadata tags of the item. To reduce misclassification, we also used a deterministic keyword-based fallback for items with uncertain template assignments, like in domains such as biostatistics and epidemiology. This hybrid approach significantly reduces noise in specialized domains where purely semantic vector matching may struggle to distinguish between subtle sub-competencies.

4.2 Hierarchical Dynamic Few-Shot Retrieval

Because LLMs outputs can be sensitive to the choice of in-context examples, fixed few-shot prompts may be poorly aligned when a query item differs in subdomain, competency, or explanation format. Prior work has shown that retrieval and example selection can substantially affect downstream LLM performance. To address this, we dynamically retrieve a small set of exemplars for each query item q from a reference pool $\mathcal{D}_{ref}$. Rather

than relying only on semantic similarity, we rank candidates using a hierarchical relevance scheme that first prioritizes metadata alignment and then uses textual overlap as a secondary signal.

Specifically, the ranking favors exact matches on the full Content Area code, then broader partial matches on the Content Area prefix, followed by matches on Competency Group and Template type. Lexical overlap between the query and candidate item is used to refine the ordering among otherwise suitable examples. This strategy encourages the retrieved exemplars to match not only the topic of the item but also its instructional framing and expected explanation style. We then insert the top- k examples (with $k = 3$) into the prompt, yielding a query-specific few-shot context that is more stable and better aligned than a static prompt.

4.3 Iterative Optimization and Generation

The generation phase employs GPT-5 as the backbone reasoning engine. Prior to large-scale inference, we perform an Expert-in-the-Loop calibration phase to align the model’s output with the specific pedagogical voice of the assessment. This process treats prompt engineering as a discrete optimization problem (Pryzant et al., 2023), where the objective function is the satisfaction score of a human Subject Matter Expert (SME). In our case, the SME was an editor expert with no clinical background but with experience in editing MCQ explanations for clarity and consistency. In each iteration t , the system generates a batch of explanations R_t using prompt P_t . The SME evaluates these outputs for consistency, format, style guidelines, and presentation logic, providing unstructured natural language feedback F_t . We utilize a meta-prompting strategy to inject this feedback into the prompt instructions, deriving a refined prompt P_{t+1} . This iterative cycle continues until the SME confirms that the outputs consistently meets the required standard without further intervention, ensuring the instructions generalize well to unseen data (Shah, 2025). Following prompt calibration, the system executes the inference pipeline. To mitigate the risk of hallucinations—a critical failure mode in medical QA—we implement a deterministic self-correction loop. The system parses the generated output to verify that every distractor option is explicitly addressed and logically refuted. If the parser detects missing components or structural inconsistencies, an automated “nudge” prompt triggers a regeneration pass. This iterative refinement

helps make the final output more structurally complete and better aligned with the target explanation style (Madaan et al., 2023). Medical consistency was assessed separately through expert review and the blinded comparative study.

5 Study Design and Findings

We evaluated the 50 AI-generated explanations through an external rating study by comparing them to the 50 existing expert-written explanations for corresponding MCQs in Set 2.

The study used a balanced incomplete block design implemented in Qualtrics. For each Set 2 MCQ, participants evaluated two explanations: one AI-generated explanation and one expert-written explanation. The source of each explanation was blinded. Each participant rated explanations for 25 of the 50 MCQs. Participants included 8 medical faculty members and 16 third-year medical students. In the faculty group, each MCQ was evaluated by four faculty members; in the student group, each MCQ was evaluated by seven to nine students.

After reading an explanation, participants rated its *clarity*, *amount of information*, and *structure* on a three-point scale (Poor, Fair, Good). Faculty also indicated whether an explanation needed correction, and both groups could provide optional open-ended comments. For inferential analysis, we fit separate cumulative link mixed models for the three rated dimensions with fixed effects for explanation type, rater group, and their interaction, plus random intercepts for rater and question.

Table 1 shows that average ratings were high for both sources, especially for clarity and structure, where scores clustered near the upper end of the three-point scale. This ceiling effect helps explain why source differences on those dimensions remained small. Even so, AI explanations received slightly higher mean ratings than expert-written explanations in all six source-by-group comparisons. Median-based summaries pointed in the same direction (see Table 2).

The clearest advantage for AI appeared in the *amount-of-information* dimension. A cumulative link mixed model estimated that AI explanations were nearly twice as likely as expert-written explanations to receive a higher rating (OR = 1.99, 95% CI [1.33, 2.99], $p = 0.001$). Neither rater group nor the interaction between rater group and explanation type was statistically significant, suggesting

that faculty and students did not differ systematically in their rating patterns and that the relative advantage of AI explanations was similar across both groups.

Lastly, faculty flagged fewer AI explanations than expert-written explanations for needing corrections: 10 of 50 AI explanations (20%) versus 19 of 50 expert-written explanations (38%). Most flagged cases were identified by only one reviewer, suggesting subjectivity.

6 Conclusion

This study demonstrates that a retrieval-guided LLM pipeline can generate explanations for MCQs that are comparable to expert-authored explanations under blinded human evaluation. Across both faculty and student raters, AI explanations matched expert explanations on clarity and structure and were rated significantly higher on the amount-of-information dimension. These results suggest that LLM-assisted workflows can meaningfully support the production of MCQ explanations in medicine. A strength of this study is its inclusion of both clinical faculty and medical students, allowing explanation quality to be assessed from the perspectives of expert reviewers and intended learners.

However, even highly rated AI explanations occasionally included verbosity, stylistic inconsistencies, or unsupported clinical details, necessitating expert review. Rather than investing effort in drafting explanations from scratch, experts can focus on review, calibration, and targeted revision to ensure clinical accuracy and pedagogical quality. More broadly, these results point to a practical role for AI in assessment development pipelines: accelerating the generation of high-quality first drafts while preserving expert oversight as the central mechanism for maintaining reliability and trust.

7 Limitations and Ethical Considerations

This study has several limitations. First, the evaluation was conducted on a single set of medical self-assessment explanations from one assessment context and therefore reflects one content domain and one editorial style rather than a fuller range of medical assessment materials. Second, the external review sample was modest (8 faculty and 16 third-year medical students, all U.S.-based), and each participant rated only a subset of explanation pairs under a balanced incomplete block design. Third, the three-point Poor/Fair/Good scale reduced re-

Dimension	Students		Faculty	
	Expert	AI	Expert	AI
Clarity	2.82 (0.44)	2.85 (0.41)	2.79 (0.46)	2.86 (0.38)
Amount of information	2.32 (0.70)	2.50 (0.66)	2.46 (0.57)	2.62 (0.63)
Structure	2.78 (0.46)	2.81 (0.44)	2.75 (0.51)	2.82 (0.41)

Table 1: Mean (SD) ratings by rater group and explanation source.

Dimension	Students		Faculty	
	Expert	AI	Expert	AI
Clarity	3 (0)	3 (0)	3 (0)	3 (0)
Amount of information	2 (1)	3 (1)	2 (1)	3 (1)
Structure	3 (0)	3 (0)	3 (0)	3 (0)

Table 2: Median and interquartile range of medians across explanations.

spondent burden for a lengthy rating task, but it also compressed variation and likely contributed to ceiling effects on clarity and structure.

From an ethical and operational perspective, we frame this work as AI-assisted content generation within a human-reviewed editorial process, not as replacement of human expertise. In this study, generated explanations remained subject to editor screening and expert review for clarity, alignment with the vignette, medical consistency, and bias. The need for this oversight is underscored by observed issues such as redundancy, imprecise wording, and occasional content misrepresentation in generated outputs. Because the study involves proprietary assessment materials and externally authored comparison explanations, we intentionally abstract some operational details and refer to external contributors generically. More broadly, the goal of this work is not to suggest that explanation authoring can be fully automated, but to evaluate whether retrieval-guided generation can support a more efficient and responsible human-centered authoring workflow for educational content.

Bias is another important reason for human oversight. Language models trained on large corpora may reflect biases related to patient demographics, disease prevalence, or clinical assumptions. If such biases appear in explanations, they could reinforce stereotypes or present an unbalanced view of clinical practice. Systematic evaluation for bias and careful editorial review are therefore necessary components of responsible deployment.

Transparency and accountability are also important in assessment contexts. Organizations using AI-assisted workflows must determine how to communicate the role of AI in content development and ensure that responsibility for the final content remains clearly assigned to human experts. More

broadly, AI systems should be positioned as tools that support expert authorship rather than replace it, helping to reduce drafting effort while preserving expert oversight and responsibility for educational quality.

References

- Iñigo Alonso, Maite Oronoz, and Rodrigo Agerri. 2024. Medexpqa: Multilingual benchmarking of large language models for medical question answering. *Artificial intelligence in medicine*, 155:102938.
- Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. 2025. Benchmarking large language models on answering and explaining challenging medical questions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3563–3599.
- Peter Y Ch'en, Wesley Day, Ryan C Pekson, Juan Barrientos, William B Burton, Allison B Ludwig, Sunit P Jariwala, and Todd Cassese. 2025. Gpt-4 generated answer rationales to multiple choice assessment questions in undergraduate medical education. *BMC Medical Education*, 25(1):333.
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.
- Yunsoo Kim, Jinge Wu, Yusuf Abdulle, and Honghan Wu. 2024. Medexqa: Medical question answering benchmark with multiple explanations. In *Proceedings of the 23rd Workshop on biomedical natural language processing*, pages 167–181.

- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.
- Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.
- Chirag Shah. 2025. From prompt engineering to prompt science with humans in the loop. *Communications of the ACM*, 68(6):54–61.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950.
- Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767.

A Sample Item and Explanation

A.1 Item

A 58-year-old woman is brought to the emergency department 4 hours after the sudden onset of left-sided facial, arm, and leg weakness. She has hypertension treated with a diuretic. She reports the sensation of a fluttering heart for 1 week. Her pulse is 125/min and irregular, and blood pressure is 135/80 mm Hg. Physical examination shows a left-sided facial droop. There is flaccid weakness of the left upper and lower extremities. An ECG shows a rapid, irregular ventricular rate with no discernible P waves. Which of the following therapies would best improve function of the weakened muscles over the next 3 hours?

- A. Acetylcholine agonist
- B. Acetylcholine antagonist
- C. Antifibrinolytic
- D. Fibrinolytic
- E. Procoagulant

Correct Answer. D.

A.2 Explanation

This patient with a likely thromboembolic cerebral infarction (stroke) from atrial fibrillation would benefit from fibrinolytic therapy. Strokes occur due to ischemic or hemorrhagic loss of blood supply to the brain and manifest as focal neurological deficits related to the dysfunction of the affected brain region. Approximately 80–85% of strokes are ischemic, commonly arising from thromboembolic disease, for which atrial fibrillation is a major risk factor. Atrial fibrillation typically presents with palpitations, fatigue, lightheadedness, and mild dyspnea if symptomatic. On ECG, atrial fibrillation demonstrates irregularly irregular RR intervals without discernible P waves. Prolonged atrial fibrillation leads to left atrial hemostasis and increases the risk of thrombosis. A thrombus from this patient's left atrium likely embolized to the right internal carotid artery supplying the right precentral gyrus, resulting in left-sided hemiparesis. Alteplase (tissue plasminogen activator) is a fibrinolytic medication utilized in ischemic strokes that binds to fibrin clots and converts plasminogen to plasmin, which lyses clots. If given early (generally within 3–4.5 hours after onset of the event), alteplase may promote neurological recovery.

Incorrect Answers: A, B, C, and E.

Acetylcholine agonists (Choice A) and antagonists (Choice B) are unlikely to be helpful in lysing this patient's thromboembolic clot, as the interaction of acetylcholine with the coagulation cascade is poorly defined. Acetylcholine agonists (eg, bethanechol) may improve cognition in patients with dementia, decrease heart rate, improve gut peristalsis, increase bladder contraction, and increase exocrine gland secretions. Acetylcholine antagonists (eg, benztropine) typically increase heart rate, decrease gut and bladder activity, and worsen cognitive function. They also act on muscle at the motor end-plate; however, this patient's weakness results from central nervous system dysfunction without pathology at the muscle fiber itself.

Antifibrinolytic (Choice C) therapy (eg, tranexamic acid) displaces plasminogen from fibrin clots to promote hemostasis during intraoperative bleeding, heavy menstrual bleeding, and traumatic hemorrhage. Procoagulant (Choice E) therapy (eg, protamine, coagulation factors) increases activation of the coagulation cascade. Both antifibrinolytic and procoagulant therapy would not lyse this patient's clot and may lead to further thrombosis.

Educational Objective. Patients with atrial fibrillation are at risk for thromboembolism due to left atrial hemostasis, which may result in ischemic strokes. Strokes manifest as focal neurological deficits related to the dysfunction of the affected brain region. Ischemic strokes are treated with fibrinolytic therapy, which promotes neurological recovery.

Sample item and explanation © National Board of Medical Examiners. All Rights Reserved. Used with permission.

B Additional Statistical Tables

Dimension	Source	Students			Faculty		
		Poor	Fair	Good	Poor	Fair	Good
Clarity	Expert	2.0	14.5	83.5	2.0	17.5	80.5
	AI	2.0	11.2	86.8	1.0	12.5	86.5
Amount of information	Expert	13.8	40.5	45.8	4.0	46.5	49.5
	AI	9.0	32.5	58.5	8.0	22.5	69.5
Structure	Expert	1.8	18.8	79.5	3.5	18.0	78.5
	AI	2.0	15.0	83.0	1.0	16.5	82.5

Table 3: Percentage of explanations by median rating, dimension, and source.

Predictor	Odds Ratio	95% CI	<i>p</i>
Explanation type [AI]	1.64	0.94–2.86	0.084
Rater group [Student]	1.22	0.50–2.99	0.669
Type $\times$ rater group	0.80	0.40–1.59	0.525
τ_{00} Questions		0.05	
τ_{00} Raters		0.79	
Marginal / Conditional R^2		0.008 / 0.209	

Table 4: Cumulative link mixed model for clarity.

Predictor	Odds Ratio	95% CI	<i>p</i>
Explanation type [AI]	1.99	1.33–2.99	0.001
Rater group [Student]	0.73	0.43–1.24	0.242
Type $\times$ rater group	0.84	0.51–1.37	0.478
τ_{00} Questions		0.00	
τ_{00} Raters		0.24	
Marginal / Conditional R^2		0.035 / NA	

Table 5: Cumulative link mixed model for amount of information.

Predictor	Odds Ratio	95% CI	<i>p</i>
Explanation type [AI]	1.48	0.85–2.57	0.168
Rater group [Student]	0.96	0.32–2.84	0.937
Type $\times$ rater group	0.86	0.44–1.69	0.670
τ_{00} Questions		0.19	
τ_{00} Raters		1.27	
Marginal / Conditional R^2		0.005 / 0.312	

Table 6: Cumulative link mixed model for Structure.

Error pattern	Questions	Percentage
No errors in either explanation	26	52%
Only expert explanation had errors	14	28%
Only AI explanation had errors	5	10%
Both explanations had errors	5	10%
Total	50	100%

Table 7: Question-level error pattern based on faculty flags.

What Aggregate Scores Hide: Per-Rule Evaluation of Russian Grammatical Error Correction

Anna Smirnova^{1*} Artyom Kopan² Vladislav Makeev² George Chernishev²

¹HEC Lausanne, University of Lausanne ²Saint Petersburg State University

anna.smirnova@unil.ch artyom.kopan@gmail.com

makeev.vladislav.d@gmail.com chernishev@gmail.com

Abstract

Russian grammar correction models can improve on aggregate benchmarks while getting worse at specific grammar rules. We show this through per-rule evaluation on a diagnostic benchmark of 48 prescriptive rules: finetuning on synthetic data improves overall $F_{0.5}$ while driving subordinate-clause comma accuracy from 14% to 1%. The suppression is invisible under corpus-level metrics and undetectable with existing coarse, corpus-specific tagsets; it is recoverable only when diagnosed at rule granularity. To enable this analysis, we develop a 98-category error taxonomy grounded in Rozental’s reference grammar and SyntErr, an open-source synthetic data generator whose per-rule distribution is an explicit parameter, designed to support arbitrary rule sets and languages. Finetuning eight open models (0.8B–12B) on 39K synthetic examples yields up to 75.3 $F_{0.5}$, approaching frontier API models with models small enough to run on device. We release the taxonomy, generator, per-rule evaluation data, and all training artifacts.

1 Introduction

Russian grammatical error correction remains a small field with limited resources: the largest annotated corpus contains fewer than 15K sentences, and each of the handful of existing benchmarks defines its own error categories and tagset. These tagsets serve different purposes: RLC (Kosakin et al., 2024) classifies errors by their likely cause in the writer (e.g., case-government errors after a verb or preposition; transfer from the writer’s first language, L1), which is useful for second-language (L2) pedagogy; RULEC-GEC and GERA use POS-based categories suited to automatic annotation. But neither approach identifies which prescriptive rule was violated.

Recent work has shown that generative LLMs are increasingly competitive with sequence label-

ing models for GEC, but they are still evaluated with the same aggregate metrics. This is problematic because the failure modes differ. Sequence-labeling models make one wrong prediction at a time, on individual tokens, which is easy to inspect. Generative models can systematically fail to apply a rule that was underrepresented in their training data, a failure that no aggregate score will reveal. Diagnosing the latter requires mapping errors to specific grammar rules.

We cross-reference three Russian GEC corpora against Rozental’s reference grammar (Rozental’, 1997), the most widely used prescriptive handbook for Russian writing norms¹, covering orthography, punctuation, and morphology/stylistics across 213 paragraphs. Coverage is sparse: the best corpus covers 65% of paragraphs, and 47 paragraphs have no attestation in any corpus.

A deeper problem is the data itself. RULEC-GEC and RU-Lang8, both L2 corpora, remain the primary training resources for Russian GEC, so current systems are optimized for learner errors. But L1 and L2 error distributions barely overlap outside of spelling: L2 corpora contain virtually no prescriptive rule violations, and some L2 errors superficially resemble L1 errors without sharing the same underlying cause. It has been shown (Sorokin and Nasyrova, 2025) that training on L2 data actively degrades performance on L1 rule-specific correction.

Our central argument is that GEC for morphologically rich languages suffers from three entangled problems that current approaches fail to decompose: (1) the sequence labeling formulation conflates rule identity, corpus frequency, and the specific replacement word into a single classification decision; (2) training on L2 corpora actively

¹For orthography and punctuation alone, the RAS-endorsed handbook by Lopatin et al. has greater institutional authority, but does not cover morphological norms, government, or stylistics, which are all essential for GEC.

* Corresponding author.

suppresses L1 prescriptive rules because the two error populations are largely disjoint; (3) aggregate evaluation hides rule-level damage, making both problems invisible. We address all three through rule-grounded synthetic data (SyntErr), per-rule evaluation anchored in Rozental’s reference grammar, and per-rule diagnostics that reveal training-data-induced suppression.

Our contributions are:

- **Taxonomy and cross-reference.** A three-level error taxonomy (8 categories / 29 sub-categories / 98 fine-grained tags) grounded in Rozental’s 213 paragraphs. We cross-reference three corpora (GERA, RULEC-GEC, LORuGEC) against this taxonomy and provide an empirical classification of 4,682 of GERA’s 5,988 errors into our categories via LLM-assisted classification (Claude Sonnet 4.6), revealing per-corpus coverage gaps (§3).
- **SyntErr.** An open-source rule-based error generator that ties each synthetic error to a specific Rozental paragraph, with configurable per-rule error rates. The current release covers the 48 rules tested by LORuGEC (Sorokin and Nasyrova, 2025), a subset of our 98 categories that future work can extend.
- **Experimental evidence.** Fine-tuning eight open models (0.8B–12B) from three families on 39K SyntErr examples plus 348 LORuGEC examples reaches 75.3 $F_{0.5}$, approaching frontier APIs with on-device-sized models (§5).

2 Related Work

We review existing Russian GEC corpora (§2.1), synthetic data generation methods (§2.2), and position our approach relative to both.

2.1 Corpora

Table 1 summarizes the publicly available Russian GEC corpora. Tag-set granularity varies by more than an order of magnitude, from 4 coarse edit labels in RU-Lang8 to 57 cause-oriented tags in RLC-GEC. Most are L2 (learner) corpora; only GERA, LORuGEC, and CoRST contain L1 errors, and LORuGEC is the only one annotated at the level of individual grammar rules. Two of the larger resources (RLC, CoRST) are accessible only through

a query interface rather than as bulk downloads. No corpus is annotated against a reference grammar.

RULEC-GEC (Rozovskaya and Roth, 2019), the main Russian GEC benchmark, contains 12.5K L2/heritage sentences annotated with 23 morphosyntactic tags. RU-Lang8 adds L2 author diversity but uses only four coarse edit labels. RLC-GEC (Kosakin et al., 2024) is a fully annotated subset of the Russian Learner Corpus with 57 cause-oriented tags. LORuGEC (Sorokin and Nasyrova, 2025) is the first Russian corpus annotated at the level of individual grammar rules (48 rules, 960 sentences).

Beyond error-annotated corpora, RuBLiMP (Taktasheva et al., 2024) provides 45K minimal pairs isolating 45 grammatical phenomena, showing that even large LMs fail on negation, reflexivity, tense, and aspect. RuCoLA (Mikhailov et al., 2022) offers 9.8K sentences for binary acceptability judgments. We use both as supplementary evaluation (§5).

2.2 Synthetic Data and Rule-Based Evaluation

Tagged corruption models (Stahlberg and Kumar, 2021, 2024) train a neural network to introduce errors guided by error type labels, but for Russian the only available labels come from 5K RULEC-GEC sentences with a coarse 20-tag set, so the generator reproduces L2 patterns and cannot generate L1 rule violations absent from its training data. Rule-based generation sidesteps this: GECTurk (Kara et al., 2023) (Turkish), Ma et al. (2022) (native Chinese GEC, closest to ours in motivation), Palma Gomez et al. (2023) (Ukrainian), and Náplava and Straka (2019); Náplava et al. (2022) (Czech) all define language-specific heuristics. Two recent studies anchor evaluation in reference grammars: Li et al. (2026) probe LLMs against 673 points from a Luxembourgish grammar book, finding that translation quality does not predict grammatical competence; Banno et al. (2026) use the English Grammar Profile to detect constructs in L2 writing. Sorokin and Nasyrova (2025) improve Russian GEC via retrieval-augmented prompting (41.0 $\rightarrow$ 56.1 $F_{0.5}$ on LORuGEC).

Across these resources and methods, three gaps recur. No Russian GEC corpus annotates errors against a reference grammar, so coverage of prescriptive rules is unknown. No Russian generator offers explicit per-rule control over the error distribution it produces; existing generators inherit their

Table 1: GEC corpora with natural Russian-language data

Corpus	Year	Size (annotated / full)	Source	Tag set size	Format
RLC-GEC	2024	31,519	L2	57	CSV
RLC	2013	~90,000 / 193,189	L2/HL	46+	Query interface
RULEC-GEC	2019	12,480	L2/HL	23	M2
GERA	2024	6,681	L1, schoolchildren	31	M2
LORuGEC	2025	960	L1 grammar resources	48	M2
CoRST	2013	~20,000 / ~175,000	L1, students	39	Query interface
RU-Lang8	2021	4,412 / 51,575	L2	4	M2

distribution from the L2 corpora they were trained on. And evaluation reports single aggregate metrics that cannot separate gains in frequent rules from losses in rare ones. The rest of the paper takes up these three gaps in turn.

3 Analysis

3.1 Taxonomy

We anchor our error taxonomy in Rozental’s Справочник по правописанию и стилистике (Rozental’, 1997) (Handbook on Spelling and Style), the most widely used prescriptive handbook for Russian. It comprises three parts (orthography §§ 1–74, punctuation §§ 75–138, morphology/stylistics §§ 139–213) and consists of 213 paragraphs. We define a three-level taxonomy: 8 coarse categories (e.g., SPELL, PUNCT), 29 sub-categories, and 98 fine-grained tags. Each fine-grained tag maps to one or more Rozental paragraphs and carries an applicability rating (full, partial, or L1-only). Parts I–II describe hard rules violated by both L1 and L2 writers. Part III is different: it covers morphological variants (e.g., § 198: в отпуске vs. the colloquial в отпуску), verbal government, agreement subtleties, and stylistic norms — prescriptive preferences that L2 learners rarely encounter. Of our 98 fine-grained tags, 55 are fully applicable to L2 writers, 28 partially, and 15 are L1-only. This makes the same taxonomy usable for evaluating both native-speaker and learner GEC systems.

3.2 Coverage Analysis

Existing corpora follow a corpus-driven error-set approach: tags are derived from errors already present in the data. The resulting tagsets are coarse. GERA’s single PUNCT tag spans 16 distinct punctuation rules in our taxonomy; S:ORTH covers 24 spelling subcategories (Table 3). A model that corrects only doubled consonants but fails on root

vowel alternations is indistinguishable under these tags. Prescriptive rules, by contrast, are explicit, reproducible, and citable: when a model fails on Rozental § 87 (comma in compound sentences), the failure traces to a specific norm. A full cross-reference (supplementary materials) confirms the scale of the mismatch: 77 of our 98 fine-grained tags fall under just 20 RULEC tags, and 95 fall under just 27 GERA tags.

To assess coverage at the level of individual rules, we built a per-paragraph evidence table² for three corpora: two L1 (GERA, LORuGEC) and one L2 (RULEC). The pipeline uses the Claude Sonnet 4.6 API: for each Rozental paragraph it searches the corpus for candidate error examples, classifies them to the specific sub-rule, and records the result. For GERA (5,988 errors) we additionally classified each error into our fine-grained tags, grouping errors by their original GERA tag so each classification request covered one tag at a time (Table 3). 4,682 errors received a fine-grained tag; the remaining 1,306 fell outside the 48 currently-implemented rules or the LLM declined to classify them. All assignments were reviewed by a single annotator (a native Russian speaker); approximately 12% of the model’s initial assignments were corrected during review.

Table 2 shows that coverage is incomplete even at the paragraph level. Within each paragraph the situation is worse. A single Rozental paragraph may contain up to twenty sub-rules (the general rule, its exceptions, and special cases), but corpora typically attest only one of them, and sometimes only an exception rather than the general rule it modifies. The mapping is not clean in either direction: one correction can correspond to several rules, and one sentence can contain several distinct errors. RULEC has no punctuation coverage at all,

²Supplementary materials: <https://synterr-nlp.github.io/papers/bea-2026/>.

	GERA	RULEC	LOR	Any
I. Orth. (74 §§)	59	54	24	66
II. Punct. (64 §§)	41	0	11	42
III. Morph. (75 §§)	39	41	7	58
Total (213)	139	95	42	166

Table 2: Evidence coverage: Rozental paragraphs with at least one verified error example per corpus. RULEC has zero punctuation coverage. 47 paragraphs are untested in any corpus. Condensed by Rozental chapter; the full per-paragraph table (213 rows) is in the supplementary materials.

GERA tag	N	N _{tags}	Top categories
PUNCT	2542	16	subordinate cl. (668), parenthetical (263), isolation (239)
S:ORTH	975	24	checked root (145), unchecked root (124), conjunction sp. (101)
G:NOUN:CASE	303	10	double gov. (110), prep. case (63)
S:LETTER:CASE	244	4	proper nouns (117), titles (63)
L:OTHER	233	3	paronyms (212)
L:REP	286	2	pleonasm (249)

Table 3: GERA error reclassification. Each GERA tag maps to multiple fine-grained categories in our 98-tag Rozental-grounded taxonomy. Automated classification by Claude Sonnet 4.6 with errors grouped by their original GERA tag so each request covered one tag at a time; results manually verified. N_{tags}: number of distinct fine-grained tags per GERA tag. GERA’s single PUNCT tag spans 16 distinct punctuation rules; S:ORTH spans 24 spelling subcategories. Full distribution available at <https://synterr-nlp.github.io/papers/bea-2026/>.

and across all three corpora 47 paragraphs (about a fifth of Rozental) have no attestation.

The gaps are not random. Learner corpora (RULEC, RU-Lang8) systematically miss the L1-only rules, which is why training on L2 data has been shown to degrade L1 correction performance (Sorokin and Nasyrova, 2025). These gaps propagate directly into training data: a model trained on a corpus that lacks punctuation examples will not learn punctuation rules, regardless of architecture. With the shift toward LLM-based GEC this bottleneck becomes addressable. LLMs already command Russian grammar from pretraining; what they need is a rule-level training signal, not token rewrites, to improve on specific prescriptive norms.

3.3 Why Generative Models?

The sequence labeling formulation of GEC (classifying each token as KEEP, DELETE, REPLACE, or APPEND (Omelianchuk et al., 2020)) has been the dominant paradigm for six years: it guarantees minimal edits by construction, producing corrections that preserve the author’s text rather than rewriting it. For English, where most corrections are single-token edits on a language with minimal inflection, the label vocabulary stays manageable.

For morphologically rich languages like Russian, the formulation entangles three things: the linguistic rule being violated, the frequency of that violation in the training corpus, and the specific replacement word. For example, the same case error gets a different label when it appears on книга than on стол: GECToR-style labels are concrete output forms (e.g., \$REPLACE_книгу), not abstract operations like “put in the genitive,” so each word × case combination demands its own label. GECToR-style models pick each token’s correction from a fixed vocabulary of 5–20K discrete labels. This vocabulary grows combinatorially with case, number, gender, and animacy, and cannot generalize to rule–word combinations absent from training.

This class imbalance is severe: in both RULEC-GEC and GERA, 94% of source tokens are already correct and receive the KEEP label during training. A rule that appears in a handful of training examples can’t compete with this signal: the model defaults to copying rather than correcting. This explains the finding of Sorokin and Nasyrova (2025) that fine-tuning on general GEC data is detrimental for rule-specific errors.

Generative models sacrifice the structural guarantee of minimal edits. In exchange, no single decision has to do all three jobs at once. The rule itself comes from pretraining. Which rules to train on comes from SyntErr, not from how often they appear in natural data. The editing behavior (*be precise, be minimal, do not rewrite*) comes from the prompt and training format, which makes it a soft constraint rather than an architectural one. Without careful prompt and data design, this softness leads to paraphrase; our overcorrection analysis (§6) quantifies this failure mode and how to mitigate it. In exchange, because the output is free text, any rule that SyntErr can generate becomes a training example with no change to the model.

4 SyntErr

We present SyntErr: an open-source error generation tool for GEC. SyntErr accepts any error taxonomy and any target distribution, extensible to new languages and rule sets. Given clean text, it introduces errors whose type, frequency, and direction are explicit YAML configuration parameters, not artifacts of corpus collection. Labels are correct by construction: each error is tagged with the rule that generated it.

Every step of the pipeline exploits linguistic information from stanza (Qi et al., 2020): POS tags and morphological features determine which tokens are valid corruption targets, dependency parses identify syntactic positions (clause boundaries for punctuation, amod arcs for agreement, obl/nmod for government targets), and pymorphy3 generates correctly inflected corrupt forms preserving all non-target features. The result is structurally realistic errors, not random noise. Concurrently, Qiu et al. (2025) reimplemented ERRANT using stanza for multilingual GEC annotation across five languages, validating stanza as the tool of choice for morphologically aware GEC pipelines.

Each example is a (src, tgt, rule) pair with one or several errors, depending on configuration. SyntErr’s minimal-pair mode (one error per sentence) is what we use for supervised fine-tuning in this paper. The current release implements all 48 rules tested by LORuGEC (Sorokin and Nasyrova, 2025), generates 10K examples in under 2 minutes on CPU, and requires no GPU or annotated data as input.

Pipeline. The generation pipeline operates in four steps. (1) An error-type distribution is specified via YAML configuration, or set to uniform (default). (2) Clean source sentences are analyzed by stanza (Qi et al., 2020), which provides both morphological tags (contextual POS, case, gender, number, e.g., стали = noun.Gen or verb.Past) and a full dependency parse tree. The parse tree is critical: it tells the generator *which* tokens stand in a syntactic relation that a given rule can target. (3) An error type is selected according to the distribution; the dependency tree is queried for applicable positions (e.g., all amod arcs for adjective–noun agreement, all clause boundaries for comma rules), and one is chosen at random. (4) The corruption is applied using pymorphy3 for morphological inflection. The output is a {src, tgt, rule} triple. The pipeline diagram is shown in Figure 2.

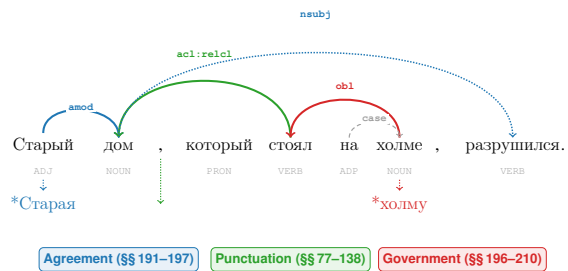


Figure 1: Dependency-driven error generation. Each arc family targets different syntactic relations: agreement (blue) uses amod/nsubj, punctuation (green) uses clause boundaries, government (red) uses obl/nmod. Corrupted forms shown below.

Most error types are not applied at random positions but at positions identified through dependency parsing: punctuation rules inspect clause boundaries and conjunction scopes, agreement rules traverse amod and nsubj arcs, and government rules use obl/nmod arcs to corrupt case frames (Figure 1). Spelling rules use dictionary lookup and Zaliznyak’s stress dictionary (Zaliznyak, 2010); inflection uses pymorphy3. Detailed rule descriptions are in Appendix D.

5 Evaluation

5.1 Experimental Setup

Models. We evaluate models in three settings: (1) frontier API models: Claude Sonnet 4.6, Claude Opus 4.5/4.6, GPT-5.4, and GigaChat-2-Max (zero-shot only); (2) eight open models fine-tuned with LoRA: Qwen 3.5 (0.8B, 4B, 9B), Qwen 2.5-7B, Gemma 3 (1B, 4B, 12B), and Apertus-8B; (3) GigaChat-3.1-Lightning (10B MoE, 1.8B active; zero-shot only). All evaluations use greedy decoding.

Data. We generated 39,209 training examples across the 48 LORuGEC evaluation rules. Source text is drawn from the RuBLiMP source pool (Taktasheva et al., 2024) (38%), Taiga (Shavrina and Shapovalova, 2017) and news corpora (Fedorov, 2019) (25%), and targeted mining for forms required by underrepresented rules (e.g., rare conjunctions, diminutive suffixes, compound adjectives; 37%). The target distribution is approximately uniform; some rules are limited by source sentence availability. All RuBLiMP benchmark sentences are excluded. Punctuation corrections are skewed 3.6:1 toward comma insertion over deletion, reflecting the natural L1 tendency to omit

required commas rather than insert spurious ones.

Fine-tuning. We fine-tune with LoRA (Hu et al., 2022) ($r=16$, $\alpha=32$, all attention and MLP projections), effective batch size 32, learning rate 10^{-4} , bf16 precision.

LORuGEC provides 348 validation and 612 test sentences. All reported scores are on the 612-sentence test set. For each model, we train three settings: (1) *SyntErr only* (39K synthetic examples), with early stopping on LORuGEC val loss (patience 5); (2) *LORuGEC only* (348 sentences — the LORuGEC val split, repurposed as training data), trained for 3 epochs; (3) *SyntErr then LORuGEC* (two-stage: first SyntErr with early stopping on val, then continued on the 348 examples for 3 epochs). Conditions (2) and (3) train on the LORuGEC val split; with 348 examples at effective batch size 32, training completes in ~ 33 steps, too few to trigger intermediate evaluation. We additionally train a clean-text control where source equals target (no errors introduced), to verify that gains come from error patterns rather than task-format exposure.

Prompt. All training and evaluation use the system prompt proposed by Sorokin and Nasyrova (2025) for LORuGEC evaluation (“Дорогая языковая модель. . .”; full text in Appendix A). This prompt instructs the model to minimally correct the input text. We adopt it to ensure fair comparison with published baselines, but note that prompt sensitivity varies substantially across model families (Table 7). We additionally evaluate frontier models with a strict prompt that explicitly forbids word substitution (“НЕ МЕНЯЙ СЛОВА” — ‘don’t change words’; full text in Appendix A). The strict prompt is used only for the prompt-sensitivity ablation (Table 7), not for the main results in Table 4: it deliberately disallows lexical substitution, which is appropriate for spelling and punctuation rules but not for morphological or stylistic ones. The baseline prompt asks for minimal correction but does so in indirect, deferential language; some models interpret this as license to paraphrase. The strict prompt constrains the model to targeted edits only. The gap reaches 16.7 $F_{0.5}$ on GigaChat-2-Max (Table 7), confirming that prompt design is a major confound in GEC evaluation (Li et al., 2026).

Benchmarks. *LORuGEC test*: 612 test sentences, 48 prescriptive rules. *GERA*: 1,314 test sentences from school texts, OOD for synthetic data. *RuCoLA*: 1,804 OOD dev sentences for gram-

maticity judgment (Matthews Correlation Coefficient; 0 = chance, 1 = perfect). All GEC scores use M2scorer $F_{0.5}$.

We normalize $\ddot{e} \rightarrow e$ and dash variants before scoring (details in Appendix E).

5.2 Results

Fine-tuning on SyntErr improves LORuGEC scores across all models (Table 4). Fine-tuning on SyntErr improves LORuGEC $F_{0.5}$ over zero-shot for every model tested. Qwen3.5-0.8B jumps from 13.7 to 45.2 (+31.5), Qwen3.5-4B from 47.6 to 67.0 (+19.4), and Apertus-8B from 48.7 to 69.4 (+20.7). Continuation on 348 real LORuGEC examples adds further gains: Qwen3.5-4B reaches 75.3 and Apertus-8B 72.4, approaching Claude Opus 4.6 zero-shot (81.8, Table 5) with far fewer parameters. For Qwen3.5-9B, SyntErr-only (57.3) still improves over zero-shot (49.2) but underperforms lorugec-only (61.2), likely due to the 3.6:1 comma insertion skew in training data; continuation training recovers to 70.9.

Clean-text control. To test whether SyntErr’s gains come from learning error patterns or simply from exposure to the prompt-and-response format, we train on clean text where the source and target are identical, following the no-corruption baseline of Kiyono et al. (2019). This collapses performance: Qwen2.5-7B drops to 2.0 on LORuGEC (from 39.5 zero-shot), Qwen3.5-0.8B to 0.0, confirming that the gains require actual error examples.

Per-rule analysis. LORuGEC’s 48 rules allow fine-grained evaluation. On Qwen3.5-4B, SyntErr→LORuGEC continuation improves 39 of 48 rules over zero-shot, degrades 5, and leaves 4 unchanged. Table 6 shows the extremes. Gains concentrate on fused/split spelling rules that SyntErr generates by construction; regressions concentrate on government and lexical rules that require semantic judgment. Notably, subordinate clause comma deletion drops from 10% to 0% under SyntErr-only: the 3.6:1 insertion skew in training data teaches the model to preserve commas rather than remove them. Continuation on 348 real examples recovers this to 50%. Full per-rule results (48 rules $\times$ 35 settings) are available at <https://synterr-nlp.github.io/papers/bea-2026/>.

Does GEC finetuning break general language understanding? Training a model to correct specific grammar rules could in principle make it reject

Model	Params	LORuGEC (612 sent.)				GERA (1,314 sent.)			
		ZS	+lor	+syn	+s→l	ZS	+lor	+syn	+s→l
<i>Prior-work baseline</i> [†]									
Qwen2.5-7B [†]	7B	39.5	44.0	58.1	66.6	47.6	44.5	45.5	50.5
<i>Qwen 3.5 family</i>									
Qwen3.5-0.8B	0.8B	13.7	33.8	45.2	54.0	25.8	38.0	44.0	36.9
Qwen3.5-4B	4B	47.6	54.7	67.0	75.3	56.5	47.4	47.9	62.8
Qwen3.5-9B	9B	49.2	61.2	57.3	70.9	56.8	54.4	52.4	59.4
<i>Gemma family</i>									
Gemma-3-1B	1B	18.9	25.5	48.0	47.8	26.6	27.6	39.3	32.2
Gemma-3-4B	4B	41.5	43.0	54.5	58.2	46.3	42.3	48.9	46.7
Gemma-3-12B	12B	52.5	58.8	66.0	69.7	53.8	55.4	59.9	57.9
<i>Other</i>									
Apertus-8B	8B	48.7	54.1	69.4	72.4	51.7	54.9	61.1	62.1
GigaChat-3.1* ^{††}	10B	52.0	—	—	—	60.4	—	—	—

Table 4: Finetuning results (M2 F_{0.5}). ZS: zero-shot (baseline prompt). +lor: LoRA on LORuGEC val (348 examples). +syn: LoRA on SyntErr (39K synthetic). +s→l: SyntErr then continued on LORuGEC val. [†]Same model used by Sorokin and Nasyrova (2025), who report 41.0 ZS / 56.1 with RAG (LORuGEC) and 67.1 finetuned (GERA). *MoE: 10B total, 1.8B active parameters. ^{††}ZS only; a 1K-token system preamble in the default chat template conflicts with the GEC instruction.

Model	LORuGEC	GERA
GigaChat-2-Max (baseline)	48.4	37.5
GigaChat-2-Max (strict)	65.1	58.7
GPT-5.4 (strict)	77.4	66.1
Claude Sonnet 4.6 (strict)	78.2	71.5
Claude Opus 4.5	79.8	64.1
Claude Opus 4.6	81.8	70.1

Table 5: Frontier API models, zero-shot (M2 F_{0.5}). Strict = explicit “do not change words” prompt.

sentences that native speakers find perfectly acceptable. To check this, we evaluate all models on RuCoLA (Mikhailov et al., 2022), a benchmark where the model judges whether Russian sentences are grammatically acceptable or not. For models ≥4B parameters, MCC remains stable across all training settings (Table 9). Smaller models (0.8B, 1B) show some degradation under SyntErr, reflecting the narrower capacity trade-off at that scale.

6 Discussion

GERA degradation. SyntErr’s source text is formal, edited prose (news, encyclopedias); GERA contains school essays by children. Qwen models show GERA degradation after SyntErr training (Qwen3.5-4B: 56.5 → 47.9), but Gemma models do not (46.3 → 48.9). The pattern is consistent across model sizes within each family, suggesting an architecture- or pretraining-level difference rather than a capacity issue. On Qwen3.5-4B, SyntErr increases the share of unchanged outputs (*copy*

Rule	ZS	+s→l	Δ
<i>Top 5 improvements</i>			
зато (spelling)	16.7	91.7	+75.0
Paired-conj. punctuation	8.3	83.3	+75.0
Dash in asyndetic sent.	6.7	80.0	+73.3
Particle -таки	7.7	76.9	+69.2
оттого (spelling)	33.3	94.4	+61.1
<i>Regressions</i>			
Prepositional case (управление)	50.0	37.5	−12.5
Suffixes -ек/-ик	55.6	44.4	−11.1
Parenthetical punctuation	57.9	47.4	−10.5
<i>Suppression + recovery (SyntErr-only → continuation)</i>			
Subord. clause commas	10.0	50.0	+40.0
(SyntErr-only: 0%)			

Table 6: Per-rule exact-match accuracy (%) on Qwen3.5-4B, LORuGEC test. ZS = zero-shot; +s→l = SyntErr then LORuGEC continuation.

in Table 8) on GERA from 7.7% to 22.5%: the model learns to be conservative from minimal-pair training and under-applies this to the noisier school-essay domain. Gemma, which starts with a lower zero-shot baseline, has less to lose from increased conservatism. Continuation on LORuGEC recovers GERA in all cases.

Prompt sensitivity. Replacing the baseline prompt with a strict variant yields up to +16.7 F_{0.5} on GigaChat-2-Max (Table 7).

Reasoning hurts. Enabling thinking mode on Qwen3.5-9B drops LORuGEC F_{0.5} from 49.2 to 23.4. The model cites Rozental by name in 165

	LORuGEC		GERA	
	Base	Strict	Base	Strict
GC-2-Max	48.4	65.1 +16.7	37.5	58.7 +21.2
GPT-5.4	73.7	77.4 +3.7	59.4	66.1 +6.7
Claude 4.6	76.7	78.2 +1.5	65.9	71.5 +5.6

Table 7: Prompt sensitivity (M2 F_{0.5}). Base = baseline prompt (Appendix A); Strict = explicit “do not change words.”

of 612 examples, but states the rule incorrectly 91% of the time. On *и так, и эдак (no comma per § 154), it finds the correct answer (“*sometimes written without commas as a fixed idiom*”) but frames it as optional and copies the input. The minimize-changes instruction interacts adversarially with the reasoning process: the model uses it to justify rejecting every correction it considers (“*since the prompt says ‘don’t change correct punctuation,’ I should not remove them*”). On rare words the model cannot recognize, reasoning produces empty output: up to 167 self-correction loops (“Wait, let me reconsider...”) consuming 14K tokens without committing to an answer. These loops are a reasoning pattern trained via reinforcement learning on math and code (DeepSeek-AI, 2025), where backtracking enables error recovery; for GEC, they amplify uncertainty rather than resolving it. The model’s prescriptive knowledge is too shallow to support productive chain-of-thought decomposition (Wei et al., 2022). The degradation is not category-specific either: under thinking, exact-match accuracy drops in every L0 category we measured (spelling −21.5, punctuation −22.4, agreement −25.6, morphology −20.0, government −56.2, lexical −10.7, averaged across the 35 LORuGEC rules mappable to our taxonomy). Mechanical categories where reasoning is plausibly unnecessary (spelling, punctuation) suffer at the same magnitude as semantic ones, ruling out the reading that thinking helps complex rules while hurting simple ones. Scaling does not help either: Qwen3.5-35B-A3B (MoE, 9B active) drops from 58.9 to 29.3 under the same conditions, confirming that reasoning-mode degradation is not a capacity issue.

The overcorrection problem. Sorokin and Nasyrova (2025) argue that generative models rewrite for fluency rather than minimally fixing errors. We quantify this directly (Table 8). On LORuGEC, zero-shot Qwen3.5-4B produces wrong edits 40.5%

	LORuGEC			GERA		
	Copy	Exact	Over	Copy	Exact	Over
ZS	31.2	28.3	40.5	7.7	40.5	51.8
+lor	5.1	23.5	71.4	3.1	13.0	83.9
+syn	38.4	38.4	23.2	22.5	55.0	22.5
+s→l	24.8	54.4	20.8	16.1	60.3	23.7

Table 8: Prediction behavior (%; Qwen3.5-4B). *Copy*: pred = source. *Exact*: pred = gold. *Over*: pred ≠ source ≠ gold. LORuGEC-only (+lor) produces 71–84% overcorrection.

of the time (overcorrection) and copies the input unchanged 31.2% (undercorrection). SyntErr finetuning halves overcorrection to 23.2% and doubles exact match from 28.3% to 38.4%. The second training stage (continuation on real LORuGEC examples) teaches the model when *not* to edit: overcorrection drops to 20.8%, exact match reaches 54.4%. LORuGEC-only finetuning (348 examples) fails badly: 71.4% overcorrection, showing the model learns to edit everything but not *what* to edit.

Per-rule analysis reveals the mechanism: on rules SyntErr covers well, overcorrection is near zero (0–6%); on rules requiring lexical judgment, it remains 43–78% as the model normalizes register rather than correcting grammar.

Aggregate scores hide rule-level damage. Averaged across all eight models, SyntErr finetuning suppresses subordinate clause comma deletion from 14% to 1%: because SyntErr generates comma *insertion* errors 3.6× more often than *deletion* errors, the model learns to preserve commas rather than remove them. Meanwhile, compound-word spelling (e.g., OTTOFO vs. OT TORO) improves from 28% to 85%. Aggregate F_{0.5} rises because the spelling gains outweigh the punctuation losses, but the punctuation rules are silently suppressed, invisible under aggregate evaluation. Continuation on 348 real examples recovers the affected rules (1% → 69%): the suppression is reversible once diagnosed. This is only possible with per-rule evaluation grounded in a reference grammar.

L1 transfer without L1 training data. Training exclusively on SyntErr (synthetic L1) and 348 LORuGEC examples transfers to GERA without any GERA-specific training: Qwen3.5-4B reaches 62.8 F_{0.5}, within 4.3 points of Sorokin and Nasyrova’s Qwen2.5-7B finetuned on GERA data (67.1). SyntErr teaches which errors to fix; LORuGEC

continuation teaches when not to edit.

7 Conclusion

We have presented three contributions toward rule-grounded GEC for morphologically rich languages. First, a 98-category taxonomy anchored in Rozenal’s reference grammar, revealing that existing corpora cover at most 65% of its paragraphs and that their coarse tagsets mask critical gaps. Second, SyntErr, an error generator whose per-rule distribution is an explicit parameter, not an artifact of corpus collection; the current release covers 48 of the taxonomy’s 98 categories and is open to extension. Third, experimental evidence across eight models from three families that 39K synthetic examples plus 348 real examples yield up to 75.3 $F_{0.5}$ on LORuGEC, approaching frontier models with 0.8B–12B open models.

Per-rule evaluation, enabled by the taxonomy, reveals what aggregate scores hide: training data can suppress specific rules (subordinate clause commas: 14% $\rightarrow$ 1%, averaged across models) while improving others (3aTo spelling: 17% $\rightarrow$ 92%), and continuation training on a small diagnostic set recovers the suppressed rules. This is the practical payoff of grounding GEC evaluation in a reference grammar.

A natural next direction is reinforcement learning rather than supervised fine-tuning. SFT remains the dominant recipe, but our results suggest it can distort the model’s edit distribution: the model learns the marginal of the synthetic data (“edit something on every sentence”) even where the input is already correct, which is exactly the overcorrection pattern we measure. A reward that captures minimal-edit behavior at the rule level should let the model retain pretrained linguistic knowledge while still picking up rule-specific corrections, without acquiring the SFT-induced bias toward editing.

We release the taxonomy with cross-reference tables, SyntErr, all training configurations and adapter weights, and per-rule evaluation data³.

Limitations

Scope. We evaluate on Russian only. We do not evaluate on L2 (learner) benchmarks or at the document level, and we use a single prompt regime in the main results (§6 reports a separate

prompt-sensitivity analysis but not a fully orthogonal sweep).

Coverage. Our taxonomy defines 98 fine-grained tags across 213 paragraphs; the current SyntErr implementation covers 48 (those tested by LORuGEC). The remaining 50 tags are implementable but not evaluated here. Punctuation generation is also directionally skewed (3.6:1 toward comma insertion; §5), which biases the kinds of punctuation errors the model ever sees during fine-tuning.

Single seed and training-stack sensitivity. We report single-seed numbers throughout and did not re-run training across multiple seeds. A small exploratory multi-seed run on two of the 4B configurations turned out to be inconclusive: the absolute scores depended substantially on the training stack (quantization, optimizer, kernel choices), and the cross-stack swing was larger than any plausible seed-to-seed variance, in a model-family-dependent way. LoRA fine-tuning at this scale is known to be sensitive to these choices, and disentangling stack effects from seed effects is beyond the scope of this paper. The absolute numbers in this paper should therefore be read with that envelope of uncertainty, even though the rule-level patterns we describe (suppression of specific rules under fine-tuning, recovery via continuation on a small diagnostic set) were consistent across all configurations we tried.

Stage order. Our two-stage models train on SyntErr first and then continue on LORuGEC. We did not test the reverse order or a randomly mixed schedule; stage order plausibly matters and is left to future work.

Validation as stopping signal. Our SyntErr-only models use the LORuGEC validation split as the early-stopping signal. This split is also used as training data in the +LORuGEC and SyntErr $\rightarrow$ LORuGEC settings, so the val-as-stop and val-as-train roles are entangled; SyntErr-only numbers may be slightly optimistic relative to a fully held-out criterion.

Single-annotator categorization. The per-paragraph mapping between Rozenal and the three corpora reflects one careful annotator’s judgment; we do not report an inter-annotator agreement score. The cross-reference should be

³All artifacts are available at <https://synterr-nlp.github.io/papers/bea-2026/>.

read as a high-effort survey rather than a validated annotation.

Strict prompt is an ablation. The *strict* prompt (§6) restricts the model to orthography and punctuation and disallows lexical substitution by design. We use it to isolate prompt sensitivity and not as a competitive system; its scope is narrower than the full GEC task.

Ethics Statement

Our taxonomy encodes prescriptive norms as a reproducible evaluation tool, not as a claim about how Russian should be written. SyntErr uses no personally identifiable data; all evaluation corpora are publicly available.

Acknowledgments

We thank the anonymous reviewers for their feedback. We thank Simon Scheidegger (HEC Lausanne) for providing access to the DGX Spark used for training and evaluation. Claude (Anthropic) was used for English prose drafting, SyntErr and evaluation pipeline development, and LLM-assisted error classification. All generated text and code were manually verified by the authors.

References

- Stefano Banno, Penny Karanasou, Kate Knill, and Mark Gales. 2026. Exploiting the English grammar profile for L2 grammatical analysis with LLMs. *arXiv preprint arXiv:2603.17171*.
- DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Maxim Fedorov. 2019. Russian keyword evaluation datasets. https://github.com/mannefedov/rw_eval_datasets.
- Gemma Team, Aishwarya Kamath, Johan Ferret, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- 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*.
- Hugging Face. 2023. PEFT: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>.
- Atakan Kara, Farrin Marouf Sofian, Andrew Bond, and Gözde Şahin. 2023. *GECTurk: Grammatical error correction and detection dataset for Turkish*. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 278–290, Nusa Dua, Bali. 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.
- Daniil Kosakin, Sergei Obiedkov, Ivan Smirnov, Ekaterina V Rakhilina, Anastasia Vyrenkova, and Ekaterina Zalivina. 2024. Russian learner corpus: Towards error-cause annotation for 12 russian. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14240–14258.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Lujun Li, Yewei Song, Lama Sleem, Yiqun Wang, Yangjie Xu, Cedric Lothritz, Niccolo Gentile, Radu State, Tegawende F. Bissyande, and Jacques Klein. 2026. *Do large language models grasp the grammar? evidence from grammar-book-guided probing in luxembourgish*. *Preprint*, arXiv:2510.24856.
- Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying Shen. 2022. *Linguistic rules-based corpus generation for native Chinese grammatical error correction*. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Vladislav Mikhailov, Tatiana Shamardina, Max Ryabinin, Alena Pestova, Ivan Smurov, and Ekaterina Artemova. 2022. *RuCoLA: Russian corpus of linguistic acceptability*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5207–5227, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jakub Náplava and Milan Straka. 2019. *Grammatical error correction in low-resource scenarios*. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

- 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.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashnyi. 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.
- Frank Palma Gomez, Alla Rozovskaya, and Dan Roth. 2023. [A low-resource approach to the grammatical error correction of Ukrainian](#). In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 114–120, Dubrovnik, Croatia. Association for Computational Linguistics.
- David Peter. 2023. hyperfine: A command-line benchmarking tool. <https://github.com/sharkdp/hyperfine>.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Jungyeul Park. 2025. Multilingual grammatical error annotation: Combining language-agnostic framework with language-specific flexibility. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*.
- Qwen Team. 2025. Qwen3.5 technical report. <https://qwenlm.github.io/blog/qwen3.5/>.
- D. E. Rozental’. 1997. *Spravochnik po pravopisaniyu i stilistike*. IC “Rossiyskaya gazeta”, Moskva. [Handbook of spelling and stylistics].
- 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.
- Sber AI. 2026. GigaChat 3.1 lightning. <https://huggingface.co/ai-sage/GigaChat3.1-10B-A1.8B-bf16>.
- Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning: “Taiga” syntax tree corpus and parser. In *Proceedings of the International Conference “Corpus Linguistics–2017”*, Saint Petersburg.
- Alexey Sorokin and Regina Nasyrova. 2025. [LLMs in alliance with edit-based models: advancing in-context learning for grammatical error correction by specific example selection](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 517–534, Vienna, Austria. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2021. [Synthetic data generation for grammatical error correction with tagged corruption models](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2024. [Synthetic data generation for low-resource grammatical error correction with tagged corruption models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 11–16, Mexico City, Mexico. Association for Computational Linguistics.
- Swiss AI. 2025. Apertus-8b-instruct-2509. <https://huggingface.co/swiss-ai/Apertus-8B-Instruct-2509>.
- Ekaterina Taktasheva, Maxim Bazhukov, Kirill Koncha, Alena Fenogenova, Ekaterina Artemova, and Vladislav Mikhailov. 2024. [RuBLiMP: Russian benchmark of linguistic minimal pairs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9268–9299, Miami, Florida, USA. Association for Computational Linguistics.
- Unsloth AI. 2024. Unsloth: Efficient LLM fine-tuning. <https://github.com/unslothai/unsloth>.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35.
- An Yang, Baosong Yang, Beichen Zhang, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- A. A. Zaliznyak. 2010. *Grammaticheskij slovar’ russkogo yazyka: Slovoizmenenie*, 6th ed. edition. AST-PRESS KNIGA, Moscow. [Grammatical Dictionary of the Russian Language: Inflection].

A Prompts

All finetuned models and baseline zero-shot evaluations use the system prompt proposed for LORuGEC evaluation:

Дорогая языковая модель, после «Исходное предложение» тебе будет дано предложение на русском языке, которое может содержать орфографические, пунктуационные, грамматические и речевые ошибки. Выведи, пожалуйста, только корректный вариант данного предложения, не давая

никаких комментариев и не выделяя никаких символов. Твоя задача — минимально изменить текст, не меняй слова и знаки препинания, которые и так правильные.

The user message template is: Исходное предложение: {src}

Strict prompt. For the prompt sensitivity experiment (Table 7), we use the following system prompt:

Исправь только знаки препинания и орфографию. Верни то же самое предложение, изменив только ошибочные запятые, тире, точки, дефисы или буквы, или их отсутствие. НЕ МЕНЯЙ СЛОВА. Каждое слово в ответе должно присутствовать в исходном предложении. Выведи только результат.

With bare {src} as the user message (no prefix).

B Training Details

Hardware. Small models ($\leq 7B$) were trained on 4× RTX 5090 (vast.ai). Large models (8–12B) were trained on a single RTX 6000 Ada (vast.ai). Evaluation was performed on a DGX Spark (GB10 Blackwell, 128GB unified memory).

LoRA configuration. All models use LoRA with $r=16$, $\alpha=32$, dropout 0, targeting all attention projections (q, k, v, o) and MLP projections ($gate, up, down$). Training uses Unsloth (Unsloth AI, 2024) with full 16-bit base weights (LoRA, not QLoRA), effective batch size 32 (per-device $4 \times$ gradient accumulation 8), learning rate 10^{-4} with cosine schedule. Early stopping monitors evaluation loss on LORuGEC validation (348 examples) with patience 5 epochs. Base training uses seed 42; continuation training uses seed 43.

Data. SyntErr training set: 39,209 examples across 48 LORuGEC rules (59 generation rules including bidirectional variants). Source text composition: 37% scarce-rule-enriched sentences (mined to boost underrepresented rules), 38% RuBLiMP scoring pool (Taktasheva et al., 2024), 25% news articles. LORuGEC training split: 348 examples (validation set of LORuGEC).

Models. Qwen3.5 (0.8B, 4B, 9B) (Qwen Team, 2025), Qwen2.5-7B (Yang et al., 2024), Gemma 3 (1B, 4B, 12B) (Gemma Team et al., 2025), Apertus-8B (Swiss AI, 2025), GigaChat-3.1-Lightning (Sber AI, 2026) (zero-shot only).

Inference. All local inference uses vLLM (Kwon et al., 2023) with bf16 precision, prefix caching enabled, and max-model-len 4096. Gemma 3 and Qwen 3.5 models require the -language-model-only flag (multimodal backbone with text-only inference). GigaChat-3.1-Lightning (Sber AI, 2026) uses the MLA attention backend with unquantized MoE routing. Frontier API models (GigaChat-2-Max, GPT-5.4, Claude Sonnet 4.6) were evaluated via their respective APIs. All models are evaluated in non-reasoning mode (-no-think): the model produces the corrected sentence directly, with no chain-of-thought or reasoning tokens.

Merge and deployment. LoRA adapters are merged into full-precision checkpoints using PEFT (Hugging Face, 2023) before evaluation. Qwen 3.5 models require AutoModelForImageTextToText for correct merging (multimodal architecture); Gemma 3 requires removal of rope_parameters from nested config levels for vLLM compatibility.

Reproducibility. All training and evaluation code, adapter checkpoints, configuration files, the SyntErr data generation pipeline, and the 39K-example SyntErr training set itself are released at <https://synterr-nlp.github.io/papers/bea-2026/>. The SyntErr source repository is at <https://github.com/synterr-nlp/synterr>. Training is deterministic given the seed, hardware, and library versions.

C Full Results

Table 10 reports full precision, recall, and $F_{0.5}$ for all models and settings on LORuGEC and GERA. Table 9 reports RuCoLA grammaticality judgment (MCC).

Model	Par.	ZS	+lor	+syn	+s→l
Qwen3.5-0.8B	0.8B	.291	.262	.166	.239
Qwen3.5-4B	4B	.460	.472	.477	.473
Qwen3.5-9B	9B	.522	.521	.521	.532
Qwen2.5-7B	7B	.493	.389	.412	.398
Gemma-3-1B	1B	.202	.211	.119	.117
Gemma-3-4B	4B	.467	.471	.414	.445
Gemma-3-12B	12B	.501	.497	.488	.520
Apertus-8B	8B	.436	.428	.479	.457

Table 9: RuCoLA OOD dev MCC (1804 examples). 0 = chance, 1 = perfect. Models $\geq 4B$: stable; smaller models degrade under SyntErr.

(a) LORuGEC													
Model	Par.	Zero-shot			+lorugec			+synterr			+s → l		
		P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}
Qwen3.5-0.8B	0.8B	21.2	5.7	13.7	45.0	16.9	33.8	60.4	22.6	45.2	60.7	37.5	54.0
Qwen3.5-4B	4B	52.3	34.9	47.6	59.8	40.8	54.7	79.7	40.9	67.0	81.8	57.1	75.3
Qwen3.5-9B	9B	51.9	40.8	49.2	65.4	48.7	61.2	62.0	44.1	57.3	74.1	60.3	70.9
Qwen2.5-7B	7B	42.0	31.9	39.5	49.4	30.6	44.0	70.6	34.0	58.1	72.3	50.6	66.6
Gemma-3-1B	1B	21.7	12.5	18.9	27.1	20.8	25.5	58.6	27.9	48.0	48.4	45.5	47.8
Gemma-3-4B	4B	43.8	34.3	41.5	43.2	42.0	43.0	59.3	41.0	54.5	59.2	54.5	58.2
Gemma-3-12B	12B	54.9	44.7	52.5	60.6	52.7	58.8	71.1	51.5	66.0	71.2	64.2	69.7
Apertus-8B	8B	51.5	39.9	48.7	66.0	31.5	54.1	78.4	47.4	69.4	76.8	58.8	72.4
GC-3.1*	10B	60.7	33.1	52.0	—	—	—	—	—	—	—	—	—

(b) GERA (out-of-domain for SyntErr)													
Model	Par.	Zero-shot			+lorugec			+synterr			+s → l		
		P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}
Qwen3.5-0.8B	0.8B	32.8	13.9	25.8	48.6	20.3	38.0	66.2	18.8	44.0	43.4	23.1	36.9
Qwen3.5-4B	4B	59.0	48.4	56.5	52.0	35.0	47.4	62.4	24.9	47.9	71.8	41.9	62.8
Qwen3.5-9B	9B	56.5	58.1	56.8	58.1	43.1	54.4	57.8	38.1	52.4	62.5	49.7	59.4
Qwen2.5-7B	7B	48.3	44.9	47.6	47.4	35.6	44.5	57.7	24.7	45.5	56.7	35.2	50.5
Gemma-3-1B	1B	27.8	22.8	26.6	27.6	27.4	27.6	50.2	21.0	39.3	32.6	30.7	32.2
Gemma-3-4B	4B	47.3	42.7	46.3	41.2	46.9	42.3	54.5	34.7	48.9	47.3	44.1	46.7
Gemma-3-12B	12B	53.6	54.6	53.8	55.2	55.8	55.4	65.3	45.2	59.9	58.7	54.8	57.9
Apertus-8B	8B	52.1	50.0	51.7	65.1	33.7	54.9	72.6	37.5	61.1	68.2	45.7	62.1
GC-3.1*	10B	65.7	45.5	60.4	—	—	—	—	—	—	—	—	—

Table 10: Full P, R, F_{0.5} on (a) LORuGEC and (b) GERA (M2scorer, ÷/dash normalized). *MoE: 10B total, 1.8B active.

D SyntErr Generation Details

Architecture. SyntErr is built around three core abstractions: ErrorHandler (a protocol from which all rule-specific error generators inherit), Analyzer (the morphological/syntactic backend), and ErrorPipeline (which orchestrates distribution sampling, candidate selection, and corruption). Rules are classified as *length-preserving* (e.g., vowel alternations, suffix errors, paronym substitution) or *length-changing* (e.g., function word insertion/deletion, compound-form splitting/merging). After any length-changing rule fires, ErrorPipeline recalculates all downstream token indices to keep the alignment between source and target consistent. The full pipeline is shown in Figure 2.

Rule examples. Consider, for example, the algorithm for introducing a "paronym" type error. First, the word is looked up in a paronym dictionary. If the word is found, it is replaced with one of its paronyms, selected at random. The new word is then inflected using pymorphy3 to preserve the original grammatical categories.

Among native speakers of Russian, spelling er-

rors involving the use of **н** and **nn** in suffixes are quite common. To simulate the error of writing **н** instead of **nn**, we utilize a morpheme dictionary to verify that **nn** constitutes part of the suffix rather than belonging to the root or appearing at a morpheme boundary. We then delete one **н**. Conversely, to generate the opposite error, we verify that **н** belongs to one of the suffixes that take a single **н** (such as **-ан-**, **-ян-**, **-ин-**) and that the target word is not among the exceptions, after which we add an additional **н**.

A more complex example involves comma placement errors (insertion of an extraneous comma). According to the rules outlined in Rozental’s reference guide, an extra comma is inserted in the following cases where it is not required:

1. A comma before the conjunction **как** when it means **в качестве** (*as*) or is part of idiom. Uses dep-tree: only targets **как** with **dep_rel=case/fixed/flat** (adjunct/apposition sense), NOT **advcl/ccomp/mark** (subordinate clause).
2. A comma within a phraseological unit containing repeated conjunctions (e.g. **ни слуху**

ни духу).

3. A comma between adjacent conjunctions at clause boundaries. Only fires when a то/так/но correlative follows the subordinate clause.
4. A comma within indivisible expressions (e.g. как ни в чём ни бывало, куда глаза глядят).

Examples of sentences with errors of these types generated using SyntErr are presented in Table 11.

Performance. SyntErr generates 10K examples in under 2 minutes on CPU; a GPU is not required but significantly accelerates stanza, especially when using depparse.

We measured generation time on a laptop (AMD Ryzen 5 7535HS, 6 cores; NVIDIA RTX 4050 6GB; 16GB RAM; Ubuntu 24.04 LTS) using a mixed source dataset compiled from RuBLiMP (Taktasheva et al., 2024), Taiga (Shavrina and Shapovalova, 2017), Russian Wikipedia,⁴ and news articles.⁵ Benchmark presets use error probability 1.0 and max one error per sentence; profile_* presets isolate handler categories.

Table 12 compares backends and depparse overhead. Natasha is fastest but lacks dependency parsing, limiting it to spelling and morphological rules. Stanza is roughly 5× slower on CPU than on GPU (Table 12); GPU throughput is roughly independent of batch size above 64 (Table 14). Per-category timings (Table 13) show uniform throughput across error types.

Rule coverage. The 48 LORuGEC rules expand to 59 generation rules (bidirectional variants for split/merge and comma insertion/deletion). Each generation rule was assigned a uniform target of 847 examples; 40 rules reached this target, and the remaining 19 are limited by source sentence availability. Punctuation rules are directionally skewed toward comma insertion (3.6:1 comma insertion skew).

Dependency-driven error generation. The key design choice is that most error types are applied at positions identified through dependency parsing, not at random. Three families of errors rely on the parse tree:

Punctuation (§§77–138): comma errors inspect clause boundaries, conjunction scopes, and modifier attachment. For compound sentences (§87), SyntErr finds the conj arc between predicates joined by и/а/но and deletes the required comma. Spurious insertion follows the same logic in reverse: inserting commas where they are *not* required (before как in non-comparative contexts, inside phraseological expressions, between adjacent conjunctions).

Agreement (§§191–197): adjective-noun errors traverse amod arcs; subject-verb errors use nsubj arcs.

Government (§§196–210): prepositional government errors corrupt case via obl/nmod arcs.

Rules not requiring syntax — spelling (§§1–34), paronym substitution, compound splitting — use morphological tags and dictionary lookup. Inflection uses pymorphy3; stress-sensitive rules use Zaliznyak’s dictionary (Zaliznyak, 2010).

Source text filtering. Before generating errors, source sentences are filtered to keep those between 10 and 50 tokens (long enough to host a target rule, short enough to validate by hand). They are then deduplicated to avoid repetition, and any sentence appearing in the RuBLiMP benchmark is removed to prevent evaluation-time leakage.

⁴<https://dumps.wikimedia.org/>

⁵https://github.com/mannefedov/ru_kw_eval_datasets

Original	Corrupted	Error	Fix tag
Отправляясь на охоту , он надел ветровку болотного цвета .	Отправляясь на охоту , он одел ветровку болотного цвета .	paronym @ position 5	\$REPLACE_надел
Шум прибоя растёт , осенний ледяной ветер вздымает и бешено сры- вает волны , разнося по воздуху брызги и резкий запах моря .	Шум прибоя растёт , осенний ледяной ветер вздымает и бешено сры- вает волны , разнося по воздуху брызги и резкий запах моря .	orthographic_spelling_nn_suffix @ position 27	\$REPLACE_ледяной
Утверждают , что бразильские карнавалы восхищают и завораживают , и когда мы впервые увидели его неповторимую яркую красоту , то сами убедились , насколько правы были очевидцы .	Утверждают , что бразильские карнавалы восхищают и завораживают , и , когда мы впервые увидели его неповторимую яркую красоту , то сами убедились , насколько правы были очевидцы .	comma_insert_ comma_between_conjunctions @ position 10	\$DELETE

Table 11: Examples of SyntErr-generated errors

Backend	Dep.	500	1K	2K	10K
<i>stanza (GPU)</i>					
+ depparse	yes	11.4	14.4	20.7	71.6
no depparse	no	9.9	12.1	16.4	51.1
<i>stanza (CPU)</i>					
+ depparse	yes	28.9	45.9	77.1	348
no depparse	no	15.9	28.2	44.4	190
natasha	no	3.4	5.4	8.7	37.3
spaCy	no	7.0	9.2	14.0	50.9

Table 12: SyntErr generation time in seconds (balanced preset; mean over 5 runs, measured with hyperfine (Peters, 2023)). lorugc preset within 5% of balanced.

Table 13: Preset performance (2K sentences, stanza GPU+depparse). S/s = sentences per second.

Preset	Time (s)	S/s	Notes
profile_spell	19.6 ± 0.2	102	no inflection
profile_morph	19.6 ± 0.1	102	pymorphy3
profile_punct	19.8 ± 0.0	101	dep traversal
profile_struct	19.5 ± 0.1	102	ins/del tokens
balanced	20.4 ± 0.7	98	reference

Table 14: Batch size vs. throughput (2K sentences). S/s = sentences per second.

Batch	Time (s)	S/s	GPU MiB
32	22.0 ± 0.1	91	1059
64	20.6 ± 0.3	97	1443
128	20.0 ± 0.1	100	1975
256	19.8 ± 0.1	101	3443
512	19.4 ± 0.2	103	3541

In force-apply mode (all 48 rules, bidirectional), stanza GPU processes 10K sentences in 79 s. Peak memory: 1.8 GB (stanza GPU+depparse), 0.4 GB (natasha), 1.1 GB (spaCy).

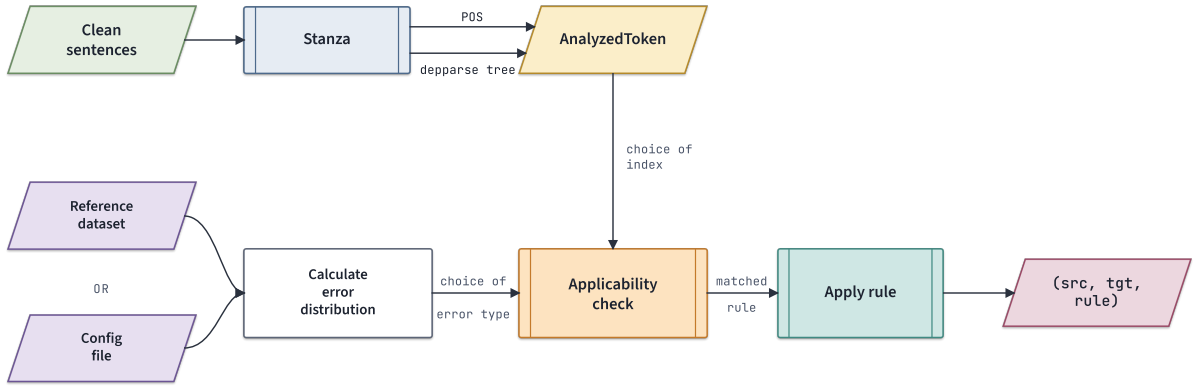


Figure 2: SyntErr error generation pipeline. Clean sentences are parsed by Stanza into analyzed tokens; a per-rule error-type distribution (from a reference corpus or a config file) drives an applicability check that selects a matched rule; the rule is then applied to produce a (src, tgt, rule) triple.

E Scoring Normalization

Before scoring, we apply two normalizations. First, `ë` is mapped to `e`: LORuGEC targets use `e` consistently, but models frequently produce the formally correct `ë`. Second, Unicode dash variants (FIGURE DASH U+2012, EN DASH U+2013) are mapped to EM DASH U+2014; the gold data uses these inconsistently (56 em-dash vs. 6 en-dash in LORuGEC). Hyphen-minus is not normalized, as it is semantically distinct. We do not normalize quotation marks: Russian prescriptive typography requires angle quotes («ёлочки»), but some gold examples use straight quotes. Models producing the correct form are penalized; we accept this small bias rather than erase a genuine distinction.

FinnGEC: Benchmarking Grammatical Error Correction for Finnish

Anh-Duc Vu,^{†‡} Mikhail Zolotilin,[‡] Jue Hou,^{†‡}

Anisia Katinskaia,[◇] Yiheng Wu,[‡] Roman Yangarber[‡]

[†]Department of Computer Science, University of Helsinki, Finland

[‡]Department of Digital Humanities, University of Helsinki, Finland

[◇]Swiss Data Science Center, EPFL & ETH Zürich, Switzerland

firstname.lastname@helsinki.fi anisia.katinskaia@epfl.ch

Abstract

Grammatical error correction (GEC) is a natural language processing task critical for improving language quality, supporting communication efficacy, and for language learning and teaching. To date, most research in GEC has focused on major, resource-rich languages such as English, while lower-resource languages remain underexplored. In this paper, we focus on GEC for Finnish. We build a dataset based on data from real-world language learners. We explore various approaches to GEC, including fine-tuning transformer models and zero-shot LLM prompting. We also adapt ERRANT, a popular GEC evaluation tool, for the Finnish language, to evaluate the performance of the models. Our results indicate that the performance of GEC for Finnish is promising, but requires further research. To the best of our knowledge, this is the first in-depth exploration of GEC for Finnish; we provide benchmarks, datasets, and code for GEC for Finnish—by releasing our training and test data and the code for Finnish ERRANT—to support further research on this important task.

1 Introduction

Grammatical error correction (GEC) is important in many areas of natural language processing (NLP), including tools for improving language quality, supporting effective communication, and language learning and teaching. Writing is a fundamental skill in second-language (L2) learning. A common approach to support L2 learners and teachers is to employ GEC systems to provide feedback to learners on samples of their writing. The objective of GEC is to identify grammatical errors, spelling errors, punctuation errors, stylistic inaccuracies, and many other types of errors, and propose corrections for them (Bryant et al., 2023). Research on GEC is decades-old, and many approaches, models, and datasets have been proposed (Bryant et al., 2019; Rothe et al., 2021; Cao et al., 2025; Deng

et al., 2025; Park et al., 2025). However, most studies to date focus on major languages, such as English, because of their rich resources, while GEC for lower-resource languages is still underexplored.

In this paper, we focus on the GEC task for Finnish, which suffers from the scarcity of GEC data: high-quality annotated data for training and testing GEC models is difficult to collect for Finnish. This complicates the training and evaluation of GEC models in Finnish. To tackle this problem, we build GEC datasets based on language-learning data from real learners. We experiment with various approaches, including fine-tuning transformer models and prompt-based methods. We adapt ERRANT (Bryant et al., 2017)—a GEC performance metric widely used, e.g., for English—to the Finnish language, and evaluate the performance of various models with our datasets. To the best of our knowledge, this is the first work that provides benchmarks for the GEC task for Finnish.

The main contributions of this paper are as follows:

- We build a Finnish GEC dataset based on real language-learning data. The dataset is based on actual errors made by real-world learners of Finnish as L2. We release this dataset to facilitate future research on GEC.
- We experiment with a number of approaches to GEC for Finnish. This provides benchmarks for future research on this task.
- We adapt ERRANT to the Finnish language and use it to evaluate model performance on our datasets. We also release the Finnish ERRANT to the research community.

The paper is organized as follows. We review related work in Section 2. We describe the sources of our data and the pre-processing steps in Section 3. We discuss our GEC methodology in Section 4. We present our experimental setup and analyze the

results in Section 5. We conclude and outline directions for future work in Section 6.

2 Related Work

GEC is typically approached as a monolingual sequence-to-sequence task, where an incorrect sentence is rewritten into its corrected version. This formulation enables the application of recurrent neural architectures, including machine translation (MT) models (Ji et al., 2017; Junczys-Dowmunt et al., 2018; Zhao et al., 2019; Rothe et al., 2021). Since the emergence of transformer models, multilingual encoder-decoder models pre-trained at scale, such as mT5 (Xue et al., 2021), have become the dominant backbone for both monolingual and multilingual GEC (Rothe et al., 2021). Given the data-intensive nature of these methods, the models are commonly pre-trained on *synthetic* datasets generated through various data-augmentation strategies, i.e., random word perturbations, Wikipedia edits, back-translation, and error-pattern imitation (Kiyono et al., 2019; Grundkiewicz et al., 2019; Takahashi et al., 2020). More recently, Wang et al. (2024) proposed contextual data augmentation to ensure more consistent error distributions in the synthetic data.

An alternative line of work formulates GEC as a sequence *tagging* task, where sentences are annotated with token transformation labels and subsequently edited based on predicted transformations (Malmi et al., 2019; Stahlberg and Kumar, 2020; Omelianchuk et al., 2020; Tarnavskiy et al., 2022). Earlier work also explored adversarial training for GEC: Raheja and Alikaniotis (2020) used a Generative Adversarial Network (GAN)-based setup, where an MT-based generator corrects errors, while a discriminator evaluates correction quality. Parnow et al. (2021) combined a sequence labeler with an error detector in a similar fashion. Li and Wang (2024) showed that explicitly decoupling error detection and correction improves performance in general language models.

Large Language Models for GEC: The emergence of LLMs has significantly impacted GEC research. GPT-3 has been shown to effectively perform GEC, generating fluent corrections that often go beyond reference corrections (Loem et al., 2023). Yet LLMs are also prone to over-editing, which can vary across languages and cause semantic shift (Katinskaia and Yangarber, 2024). Alignment-enhanced methods have been proposed

to mitigate over-corrections in both sequence-to-sequence and decoder-only LLMs, for example by training models on both good and bad corrections to instruct the model to prefer the good ones (Yang and Quan, 2024; Liang et al., 2025).

Recent work has also explored LLM-based generation of synthetic data for GEC. Luhtaru et al. (2024) fine-tuned Llama 2 models for artificial error generation, producing synthetic errors that closely resemble human-made errors and achieving state-of-the-art results on German, Ukrainian and Estonian. Potter and Yuan (2024) extended this approach to code-switched text (text mixing multiple languages) generation, addressing the challenge of correcting errors in multilingual English as a Second Language (ESL) learner writing. Koo et al. (2024) proposed Knowledge-Augmented GEC (KAGEC), incorporating external knowledge retrieval to enhance LLMs’ context-aware generation capabilities, particularly for non-English languages, including Korean.

Low-Resource and Multilingual GEC: For low-resource scenarios, Korotkova et al. (2019) investigated zero-shot GEC using Transformer-based MT models. Cross-lingual transfer has emerged as a promising direction, leveraging multilingual pre-trained representations to transfer correction capabilities across languages (Yamashita et al., 2020; Sun et al., 2022). Palma Gomez and Rozovskaya (2025) recently proposed selective data augmentation with round-trip MT (translating text into an intermediate language and back), and demonstrated substantial improvements for Russian and Ukrainian GEC. The MultiGEC-2025 shared task (Masciolini and Volodina, 2025)¹ covered 12 languages, including Estonian, the closest relative of Finnish in the Uralic language family.

Finnish GEC remains a low-resource problem with limited dedicated infrastructure. The closest related language with established GEC support is Estonian, which has recently benefited from LLM-based error generation (Luhtaru et al., 2024). For Finnish, Creutz (2024) compared GPT-3.5, GPT-4, and Claude on authentic beginner-level learner texts, finding that GPT-4 achieves the best correction performance among these models.

Evaluation and Quality Control: Automatic evaluation of GEC systems relies on gold-standard reference-based metrics computed over *edits*—

¹<https://spraakbanken.gu.se/en/compsla/multigec-2025>

words or phrases that have been changed, added, or removed when comparing the model’s output and the reference. The M2 scorer (Dahlmeier and Ng, 2012) is an early standard. It first aligns the source sentence (to be corrected) with both the model’s output and the reference to identify which tokens were changed, inserted, or deleted, and then checks how many of the model’s edits match the reference edits. ERRANT (Bryant et al., 2017) extended this approach by automatically labeling each aligned edit with an *error type*, enabling fine-grained evaluation at the level of error types, and it has become a widely accepted framework for GEC evaluation. An alternative approach avoids the alignment step by computing an N-gram F-score directly between the hypothesis and the reference without pairwise alignment, making it computationally more efficient (Koyama et al., 2024).

Kobayashi et al. (2024) demonstrated that GPT-4 achieves state-of-the-art performance as a GEC evaluator, surpassing traditional automatic metrics in correlation with human judgments. A post-processing approach has also been explored, where a classifier judges whether each proposed correction is grammatically acceptable and filters out invalid ones (Cao et al., 2024). Goto et al. (2025b) proposed a ranking-based aggregation procedure that better aligns with human judgments across metric types. To support reproducible comparisons, Goto et al. (2025a) released *gec-metrics*, a unified library implementing a wide range of GEC evaluation metrics through a consistent interface.

3 Data

3.1 Corpus of Learner Errors

We begin by building a corpus of errors from data collected from learners of Finnish as L2, who use Revita, a language learning platform (Hou et al., 2025). The platform offers learners the possibility to practice with authentic real-world texts. The platform offers multiple-choice and “cloze” (fill-in-the-blank) exercises, where the base form—*lemma*—of a word is given to the learner, and serves as a hint in the context of 2–3 sentences. We refer to the complete exercise context as a “snippet.” The learner’s task is to produce the correct form of the word that best suits the context.

We build the *learner-error* corpus by collecting *actual observed* errors from learner data. This allows us to collect a dataset with two principal advantages. First, our dataset covers multiple contexts

and error types, including subject-verb agreement, tense, preposition, noun phrase agreement, government relations, etc. Errors may appear in any position in the sentence. This is in contrast to other existing GEC datasets, such as BLiMP (Warstadt et al., 2020), which has the following limitations:

- they target only certain error types, such as Subject-Verb Agreement (SVA);
- each instance can contain only one erroneous word, and the word is tagged with only one error type;
- this also means that some of the errors appear in a fixed role in a sentence: e.g., in SVA, only the predicate can contain an error.

This means that the set of errors observable in BLiMP is severely restricted by the rules used in their collection process.

Secondly, as different learners practice with the same lemma, they commit different mistakes, including spelling, grammatical, and syntactic errors. Further, a particular mistake made by a learner may contain multiple grammatical errors; e.g., it is quite common that a learner chooses an incorrect tense, person, and number all at the same time. We believe this is a more realistic scenario, and can better test the true performance of a GEC model.

We should note that our work on GEC is driven by its applications for language teaching and language learning (Katinskaia and Yangarber, 2021, 2023; Katinskaia et al., 2019). In L2 teaching, having robust GEC is crucial for several reasons. One is to support the generation of meaningful feedback for text (e.g., essays) produced by learners. When learners perform exercises, GEC is used to detect *alternative-correct* answers, also called multiple-admissible answers. Multiple admissibility occurs when more than one form of a word fits syntactically and semantically in a given context. In such situations, it is crucial that a model can **acknowledge** the correctness of alternative answers.

The learner-error corpus contains approximately 3.7K snippets. Each snippet contains one or more errors. Overall, there are approximately 23K distinct errors from 890 distinct texts in the dataset.

The exercise types from the learning platform are cloze and multiple-choice tasks. Therefore, the learner-error corpus contains primarily morphological substitution errors. However, since some exercises involve multi-word answer slots (e.g., compound verb forms), the dataset also contains a small number of insertion and deletion errors

# of errors	Proportion in (%)	
	learner data	Wiki data
0	52.6	50.0
1	19.2	18.1
2	13.8	18.4
3	6.1	9.7
4	3.0	2.0
5	1.8	1.2
≥ 6	3.5	0.7

Table 1: Training data: proportion of snippets with different numbers of errors.

arising from word count differences between the learner response and the reference.

3.2 From Learner Errors to Instances

We next describe the process of building the datasets of *instances* for training and testing models for GEC, from the learner-error corpus. Each instance is a “parallel” pair: a snippet (sequence of words and punctuation) containing some errors, paired with a *reference*—an error-free version of the same snippet. While a snippet with errors may be corrected in *multiple ways* in real life, we provide only a single error-free reference. The reference corresponds exactly to the correct snippet found in the original text with which the learners were practicing.

We split the snippets into three sets: approximately 1K snippets reserved for testing, and the remaining 2.7K snippets split into a training set (85%) and a validation set (15%)—at the snippet level: this means that *all instances* derived from a given snippet (see below) always fall into the same split, and are never distributed across more than one split. These three splits are randomly sampled from the error corpus. To ensure that there is no leakage across the splits, we filtered the validation and test sets to remove any snippets with identical references in the training set.

Each snippet contains one or more error *slots*—positions where some learner has made a mistake. Each error slot may have multiple distinct incorrect variants—committed by different learners at different times. We generate GEC instances by combining different error slots within each snippet. For example, a snippet with two error slots, each having 4 distinct learner-error variants, yields 25 instances in total: 16 two-error instances formed by pairing each variant from the first slot with each variant from the second slot ($4 \times 4 = 16$), 8 single-error instances formed by combining each variant

from one slot with the correct form of the other ($4 + 4 = 8$), and one “clean” instance where both slots are filled with the correct form (which constitutes a *negative* instance for GEC).

For the test set, we apply this combinatorial expansion fully, to maximize the coverage of error types. When applied to the 1K test snippets, this yields approximately 70K instances. We downsample the dataset to 9,981 instances using stratified sampling. The idea is to preserve the distribution of error density—the ratio of error slots to sentence length—while also limiting the number of instances assembled from the same source snippet. This prevents snippets with more error slots from dominating over snippets with fewer error slots, ensuring a more balanced representation across source texts. Lastly, we supplement the test set with zero-error instances sampled directly from texts in the language learning platform, to evaluate accurately the over-correction rate (OC). This results in a total of 17,854 instances in the test set.

For the training and validation sets, we apply a different expansion strategy to maximize the contextual diversity for each error pattern. We first scan a snippet by its sentence boundaries and segment the snippet into overlapping windows of different sentence sizes. A window may range from one standalone sentence to multiple contiguous sentences. For example, a 3-sentence snippet would yield six possible windows: three single-sentence windows, two two-sentence windows, and one full three-sentence window. We always include the full snippet window; we randomly select another 3 from the remaining windows, so each snippet contributes at most 4 training pairs. This allows us to generate training instances that vary in size and avoid high similarity due to overlapping contexts. Too much overlap in the training data would risk that the models would try to memorize the contexts.

For each window, we identify all error slots that fall within the window and apply a set of substitutions for each slot, but not necessarily for all slots. This means only a subset of the slots may be filled with their incorrect variants, while the others remain untouched. To maintain diversity of error combinations, the same error slot, even if it appears in multiple windows, will be substituted only once with an incorrect variant across all windows of the same snippet. We expect that this approach will serve to build a more robust model.

We also include an error-free instance from each snippet. The goal is to prevent the model from

Source	Training	Validation	Test
L2 learner data	10,668	1,372	17,854 [†]
Wikipedia data	1,850,000	150,000	—

Table 2: Dataset statistics after expansion and augmentation. [†]The test set includes 9,981 error-containing instances and additional zero-error instances drawn from the platform, used to evaluate the over-correction rate.

over-correcting: proposing corrections regardless of whether an error is present. Additionally, we supplement the training and validation sets with clean zero-error instances drawn directly from the platform—following the same approach used for the test set—resulting in approximately 52.6% zero-error instances in the training data, to reduce over-correction. In total, the training set contains 10,668 pairs and the validation set contains 1,372 pairs. The distribution of errors is shown in Table 1.

We note that error-correction pairs in our data are not one-to-one: the same erroneous form may map to different correct forms depending on the context, and vice versa. The model must therefore learn context-sensitive correction rather than memorizing fixed error pairs.

3.3 Augmentation with External Corpus

Although our dataset covers a diverse range of actual error types, composing instances by error combinations would introduce a bias into the training process: the same reference snippet may appear in multiple training pairs, causing the model to overfit to specific snippet contexts rather than learning generalizable correction patterns. To mitigate this, we augment the training data with an external corpus.

We augment the training data with sentences from Finnish Wikipedia (1.2M documents), split into snippets of 1–3 consecutive sentences (8–80 words). We retain only snippets that contain words or phrases for which real learner errors exist in the learner-error corpus. For each snippet, several such words or phrases are selected at random and replaced with incorrect variants drawn from that corpus, producing erroneous training pairs. We also include a large number of clean, error-free pairs to balance the dataset and discourage the model from over-correcting. This yields 1,850,000 training and 150,000 validation pairs (Table 2).

The actual error count distribution in training is shown in Table 1.

The distribution of errors in the validation set is nearly the same. This allows us to have much more

training data with diverse contexts and references.

4 Methodology

4.1 LLM Prompting

Recent advances in LLMs have shown very strong performance on various NLP tasks. Researchers have been exploring the potential of LLMs for GEC in particular (Staruch et al., 2025; Davis et al., 2024; Katinskaia and Yangarber, 2024). We consider the prompt-based approach to be a strong baseline for the GEC task. In this work, we focus on the zero-shot setting and prompt the LLMs with only one instance at a time. The prompt is presented in Appendix B. We focus the prompt on errors in morphology and agreement, as these constitute the dominant error types in our dataset, as described in Section 3. We plan to explore the few-shot setting in future work.

In this work, we use OpenAI GPT-5.4 and Claude Haiku 4.5 as prompt-based baselines, and evaluate their performance with the full balanced test set—17,854 instances.

4.2 Fine-tuning Encoder-Decoder and Decoder-Only Models

We experiment with two encoder-decoder transformer models. **FinT5** is a 260M-parameter monolingual T5 model² pretrained exclusively on Finnish text, giving it strong morphological knowledge of Finnish. **mBART** (Tang et al., 2020) is a 611M-parameter multilingual model pretrained on 50 languages, with Finnish accounting for only a small fraction of its pretraining data. We also experiment with **Poro2** (Zosa et al., 2025), a decoder-only model pretrained on Finnish and English with 8B parameters. More specifically, we experiment with Poro2-Instruct, created through supervised fine-tuning (SFT) and Direct Preference Optimization (DPO) of the Poro 2 8B Base model.

We adopt a two-stage training process for all models to mitigate overfitting caused by repeated error combinations in our L2 learner dataset:

1. Stage 1—Fine-tune with augmented data:

We fine-tune the models on synthetic Wikipedia data with artificially injected learner errors. This aims to teach the model to perform GEC across diverse contexts.

2. Stage 2—Fine-tune with L2 learner data:

We continue fine-tuning on the L2 learner

²Finnish-NLP/t5-small-nl24-finnish

Error type	Meaning	Examples (with description)
VERB:FORM	Verb Form	sanoa → sanon (infinitive vs. finite verb)
VERB:TENSE	Tense	menee → meni (present vs. past)
VERB:VOICE	Voice	sanon → sanotaan (active vs. passive)
VERB:MOOD	Mood	tulee → tulisi (indicative vs. conditional)
VERB:PART	Participle	juokseva → juossut (present vs. past)
PERSON	Person	sanon → sanot (1 person vs. 2 person)
DEGREE	Degree	iso → isompi (positive vs. comparative)
CASE	Case	kaupunki → kaupungin (nominative vs. genitive)
NUMBER	Number	koira → koirat (singular vs. plural)
NOUN_POSS	Possessive	koira → koirani ("dog" vs. "my dog")
CLITIC	Clitic	-kin, -kaan, -han, etc. (incorrect clitics)
AGR	Agreement	(Phrase-level agreement errors)
WO	Word Order	(Word order errors)
WORD_CHOICE	Word Choice	(same POS, different lemma)
CONJ	Conjunction	ja → mutta
PART	Particle	(Particle errors)
VOWEL_HARM	Vowel Harmony	talossa → talossä
COMPOUND	Compound	autotalli → auto talli (compound joined vs. split)
SPELL	Spelling	(Typos)
PUNCT	Punctuation	! → ?
INSERT	Insertion	(Missing words)
DELETE	Deletion	(Removed words)
OTHER	Other	—

Table 3: Error types for Finnish GEC.

dataset to adapt the model to the distribution of real Finnish learner errors in practice.

For FinT5 and mBART, both stages perform full parameter fine-tuning without layer freezing, while Poro2 is trained with LoRA adapters (Hu et al., 2022) on all linear layers. In Stage 1, each model is trained for up to 2 epochs on the augmented Wikipedia data, with the best checkpoint selected according to validation loss. In Stage 2, training continues from the Stage 1 final version and uses only the learner data. Stage 2 is configured to run up to 5 epochs, and to retain the best checkpoint according to validation loss. More detailed information about the hyperparameters is available in Appendix A.

4.3 Evaluation with Finnish ERRANT

ERROR ANnotation Toolkit (ERRANT) (Bryant et al., 2017) is widely used for evaluating GEC models. It requires reference corrections, and measures the performance in terms of an edit-based F-score. Besides computing scores for the overall performance, it also evaluates the different error types. The original ERRANT is designed for English and is developed with 25 English error types. In our work, we extend ERRANT according to the Finnish grammar and common error types. We

present our error types in Table 3. We apply the Stanza parser (Qi et al., 2020) to parse the sentences and extract information about the errors.

We adapt the Finnish ERRANT with multi-label annotation. Each edit can be tagged with multiple error types simultaneously, reflecting the compounding nature of Finnish morphological errors. It is common for a Finnish learner to make a mistake with multiple errors at the same time. For example, a learner can submit the answer *sanon* ("I say," indicative mood, present tense 1st person singular), while the correct answer is *sanoi-vat* ("they said," past tense 3rd person plural). In this case, we observe three errors within one edit: tense, person, and number. This adaptation treats the evaluation as a multi-label classification problem and avoids the information loss inherent in the single-label tagging scheme as in the native ERRANT.

Before the evaluation, all texts are normalized: Unicode NFC normalization, full-width character normalization, quote character normalization, and whitespace normalization. This avoids penalizing models for surface formatting differences unrelated to grammatical correctness.

We evaluate all models using precision (P), re-

Model	Architecture	Precision	Recall	$F_{0.5}$	OC %
FinT5 (260M)	encoder-decoder	33.89	17.44	28.52	23.4
mBART (611M)	encoder-decoder	50.46	9.71	27.44	3.6
Poro2 (8B)	decoder-only	36.52	24.30	33.18	13.0

Table 4: Stage-1 performance after fine-tuning with augmented Wikipedia data. OC—over-correction rate

Model	Architecture	Precision	Recall	$F_{0.5}$	OC %
OpenAI GPT-5.4	—	47.27	60.10	49.37	57.0
Claude Haiku 4.5	—	39.23	42.31	39.81	47.9
FinT5 (260M)	encoder-decoder	48.17	40.25	46.35	23.4
mBART (611M)	encoder-decoder	51.13	22.84	40.98	9.9
Poro2 (8B)	decoder-only	40.54	43.36	41.07	22.6

Table 5: Stage-2 Performance of prompt-based (top), and fine-tuned (bottom) models after fine-tuning on L2 learner data. OC—over-correction rate.

call (R), and $F_{0.5}$:

$$F_{\beta} = \frac{(1 + \beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R} \quad (1)$$

with $\beta = 0.5$ to weight precision twice as heavily as recall, following standard GEC practice (Bryant et al., 2017). We also report the over-correction rate (OC), defined as the proportion of error-free input sentences for which the model attempts to modify something.

5 Experiments and Results

5.1 Fine-Tuning Performance

Table 4 presents the performance after Stage 1 fine-tuning on augmented Wikipedia data. Poro2 achieves the best $F_{0.5}$ of 33.18% at this stage. Among the encoder-decoder models, FinT5 achieves $F_{0.5}$ of 28.52% with an over-correction rate of 23.4%. mBART achieves the highest precision (50.46%) and the lowest over-correction rate (3.6%), suggesting it is more conservative in making corrections.

Table 5 presents the performance after Stage 2 fine-tuning on L2 learner data, alongside the prompt-based baselines. All fine-tuned models improve substantially over Stage 1. FinT5 achieves $F_{0.5}$ of 46.35% and mBART achieves 40.98%. Poro2 reaches 41.07%, slightly outperforming mBART.

Among the prompt-based approaches, GPT-5.4 achieves the best overall $F_{0.5}$ of 49.37%, outperforming all fine-tuned models, while Claude Haiku 4.5 reaches $F_{0.5}$ of 39.81%.

5.2 Error Analysis

Table 6 shows the full breakdown by $F_{0.5}$ for error types for all models. Vowel harmony errors (VOWEL:HARMONY) are the easiest to correct across all models, with all models scoring above 72%. Poro2 achieves the highest $F_{0.5}$ of 86.96%, followed by FinT5 at 83.80%. Claude Haiku has the lowest performance on the vowel harmony errors with 72.14% $F_{0.5}$.

Verb form errors (VERB:FORM) are also relatively easy to correct, with FinT5 achieving $F_{0.5}$ of 75.22% and Poro2 at 69.86%. mBART and Claude Haiku also perform competitively at 60.57% and 61.78%, suggesting that verb form correction is a tractable task for most models as it can be learned from local morphological patterns.

Case errors (CASE), on the other hand, require a deeper understanding of Finnish syntax.³ This is reflected by clear performance gaps between models. GPT-5.4 achieves the highest $F_{0.5}$ of 70.15% on case errors, followed by FinT5 at 63.09%. Claude Haiku and Poro2 both reach around 57%, while mBART reaches only 45.42%. This highlights the advantage of Finnish-specific pretraining. Both FinT5 (260M, Finnish-specific) and Poro2 (8B, Finnish and English) have clearly better performance than mBART (611M, 45.42%), the only multilingual model in the comparison, despite mBART being larger than FinT5.

Insertion, deletion, and punctuation errors (INSERT, DELETE, PUNCT) are the most challenging error types across all models, with most scoring very low on $F_{0.5}$. Unlike morphological er-

³We should note that Finnish has an extremely complex nominal case system, with at least 14 cases, possibly more, depending on the chosen theory.

Error Type	GPT-5.4	Claude Haiku	FinT5	mBART	Poro2
VOWEL:HARMONY	81.78	72.14	83.80	77.07	86.96
VERB:FORM	67.50	61.78	75.22	60.57	69.86
SPELL	69.56	60.04	48.71	55.65	54.63
CASE	70.15	57.22	63.09	45.42	57.14
DEGREE	72.89	70.75	41.79	15.12	44.03
VERB:VOICE	58.99	55.69	60.86	34.15	53.29
VERB:PART	62.88	52.84	50.70	36.67	44.64
NUMBER	61.68	50.50	57.30	44.82	52.78
CLITIC	55.56	45.59	48.08	14.85	24.76
WORD:CHOICE	54.52	42.69	33.34	27.64	41.75
VERB:TENSE	52.60	43.45	53.17	28.71	33.40
PERSON	59.56	48.32	38.55	35.71	44.08
OTHER	29.66	28.27	37.69	18.81	28.44
VERB:MOOD	27.97	27.90	22.94	28.14	18.30
COMPOUND	18.48	20.71	12.89	12.23	15.58
DELETE	7.90	6.11	2.68	5.98	3.75
INSERT	5.80	8.40	16.86	9.96	13.39
PUNCT	1.12	1.38	2.83	0.00	1.72

Table 6: $F_{0.5}$ by error type across all models.

rors, which can be corrected by modifying the word form, insertion and deletion errors require the model to determine whether a word is extraneous or missing in the context. This is also expected, as the cloze and multiple-choice exercise format produces primarily morphological substitutions, leaving the models with limited training signal for these error categories.

5.3 Manual FP Inspection

One known limitation of reference-based evaluation (Choshen et al., 2018) is that a correction is penalized if it differs from the reference, even if it is valid. As we are experimenting with a single-reference dataset, this limitation is even more pronounced. To mitigate this issue, we conducted a manual inspection of false positives (FPs) flagged by Finnish ERRANT. Due to resource constraints, we focus on four models: FinT5, mBART, Poro2, and GPT-5.4. We randomly sampled 100 instances from the intersection of FP instances produced by four models. Each FP instance may contain different edits from different models, but they all include false positives flagged by Finnish ERRANT. This reduces the workload but still allows us to inspect our model’s behavior comprehensively. A native Finnish expert then judged each instance at the sentence level, assigning either **Correct** (the model made a valid correction that ERRANT penalized) or **Wrong** (the model genuinely made an incorrect

correction).

Table 7 shows the results of the manual FP inspection. Our human evaluation shows that 51% (mBART), 61% (Poro2), 65% (FinT5), and 85% (GPT-5.4) of these flagged edits are in fact valid corrections, confirming that ERRANT underestimates true model precision.

The ranking of valid corrections reflects the overall $F_{0.5}$ ranking, suggesting that better-performing models also generate high-quality corrections even when they do not match the references. Furthermore, the large gap between GPT-5.4 (85%) and the best fine-tuned model FinT5 (65%) suggests that reference-based metrics systematically underestimate LLM performance more than fine-tuned model performance. Fine-tuned models are trained to reproduce the reference and tend to stay closer to it, while LLMs generate more diverse but potentially valid corrections.

6 Conclusion

In this paper, we present the first benchmarks for grammatical error correction for Finnish. We construct a dataset from L2 learner data. The dataset covers a diverse range of morphological and syntactic error types. We adapt ERRANT to Finnish to enable a systematic evaluation and benchmark a range of approaches spanning from prompt-based methods to fine-tuning transformer-based models.

Our results show that language-specific pre-

<i>Model</i>	<i>Valid</i>	<i>Invalid</i>
mBART (611M)	51	49
Poro2 (8B)	61	39
FinT5 (260M)	65	35
OpenAI GPT-5.4	85	15

Table 7: Manual inspection of 100 false positives flagged by ERRANT. *Valid*: the GEC model made a valid correction. *Invalid*: the GEC model made an incorrect correction.

training is a more impactful factor than model scale. FinT5 (260M parameters) outperforms the larger multilingual mBART model and outperforms Claude Haiku 4.5, a compact commercial LLM, particularly on morphologically complex error types such as case. Poro2, a decoder-only model fine-tuned under the same two-stage setup, achieves competitive results with mBART, demonstrating that decoder-only architectures can also be adapted effectively for Finnish GEC.

Finnish GEC remains an open challenge. For future research, we plan to use few-shot prompting techniques to reduce the over-correction problem and to further improve LLM-based GEC performance.

In terms of data preparation, expanding the training data to cover more error types, such as insertion, deletion, and punctuation, is an important direction. Our results show that the models currently receive insufficient training for these error categories. Scaling up training data through systematic synthetic error generation, either via LLM-based error injection or knowledge distillation from frontier models, is a potential direction for improving fine-tuned model performance.

In addition, we also plan to experiment with larger models. FinT5 already achieves competitive performance with compact commercial LLMs such as Claude Haiku 4.5 despite its small size, and experimenting with a larger Finnish-specific model may close the gap to frontier LLMs further.

Acknowledgements

This work was supported in part by the Research Council of Finland, Project “Know-AI” (Grant 1359285), and by Strategic Research Council Project “DALAI-Fin” (Grant 1373225).

Data Availability

The full training dataset, derived from learner data collected via Revita, is available upon request by

contacting the authors, subject to licensing constraints from the original data.

Limitations

Our dataset is sourced from a single language learning platform, which may limit generalization to other Finnish L2 learner populations. The raw L2 learner dataset is also relatively small, with 10,668 training, 1,372 validation, and 9,981 error-containing test pairs. Each instance provides only a single reference correction, though multiple valid corrections may exist for the same error. The dataset consists primarily of morphological substitution errors, as the cloze and multiple-choice exercise format produces mainly morphological substitutions; insertion, deletion, and punctuation errors are underrepresented, limiting the training signal available for these error categories.

Evaluation relies solely on Finnish ERRANT, an automatic metric adapted from English. For prompt-based approaches, only zero-shot prompting was evaluated, leaving few-shot settings for future work. Sequence-labeling approaches such as GECToR (Omelianchuk et al., 2020) are out of scope for this benchmark paper and left for future work.

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*, 49(3):643–701.
- Bin Cao, Kai Jiang, Fayu Pan, Chenlei Bao, and Jing Fan. 2024. [Improving grammatical error correction by correction acceptability discrimination](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8818–8827, Torino, Italia. ELRA and ICCL.
- Yayu Cao, Tianxiang Wang, Lvxiaowei Xu, Zhenyao Wang, and Ming Cai. 2025. [CxGGEC: Construction-guided grammatical error correction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6156, Vienna, Austria. Association for Computational Linguistics.
- Leshem Choshen, Lior Bar, and Omri Abend. 2018. [Inherent biases in reference-based evaluation for grammatical error correction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–641, Melbourne, Australia.
- Mathias Creutz. 2024. [Correcting challenging Finnish learner texts with claude, GPT-3.5 and GPT-4 large language models](#). In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 1–10, San Ġiljan, Malta. Association for Computational Linguistics.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.
- Jiayi Deng, Chen Chen, Chunyan Hou, and Xiaojie Yuan. 2025. [InstructGEC: Enhancing unsupervised grammatical error correction with instruction tuning](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 110–122, Abu Dhabi, UAE. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. [gec-metrics: A unified library for grammatical error correction evaluation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. [Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human?](#) In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. 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.
- Jue Hou, Anh-Duc Vu, Anisia Katinskaia, and Roman Yangarber. 2025. [AI-assisted second-language teaching and learning in the Zone of Proximal Development](#). *Studies in language assessment*, 14(2):92–122.
- 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 *ICLR*.
- Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. [A nested attention neural hybrid model for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Anisia Katinskaia, Sardana Ivanova, and Roman Yangarber. 2019. [Multiple admissibility: Judging grammaticality using unlabeled data in language learning](#). In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 12–22, Florence, Italy.
- Anisia Katinskaia and Roman Yangarber. 2021. [Assessing grammatical correctness in language learning](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 135–146, Kyiv, Ukraine.

- Anisia Katinskaia and Roman Yangarber. 2023. [Grammatical error correction for sentence-level assessment in language learning](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 488–502, Toronto, Canada.
- Anisia Katinskaia and Roman Yangarber. 2024. [GPT-3.5 for grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- 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.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Large language models are state-of-the-art evaluator for grammatical error correction](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.
- Seonmin Koo, Jinsung Kim, Chanjun Park, and Heuiseok Lim. 2024. [Search if you don’t know! knowledge-augmented Korean grammatical error correction with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 96–125, Miami, Florida, USA. Association for Computational Linguistics.
- Elizaveta Korotkova, Agnes Luhtaru, Krista Liin, Maksym Del, Daiga Deksnė, and Mark Fishel. 2019. [Grammatical error correction and style transfer via zero-shot monolingual translation](#). *Preprint*, arXiv:1903.11283.
- Shota Koyama, Ryo Nagata, Hiroya Takamura, and Naoaki Okazaki. 2024. [n-gram F-score for evaluating grammatical error correction](#). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 303–313, Tokyo, Japan. Association for Computational Linguistics.
- Wei Li and Houfeng Wang. 2024. [Detection-correction structure via general language model for grammatical error correction](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1748–1763, Bangkok, Thailand. Association for Computational Linguistics.
- Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. 2025. [Edit-wise preference optimization for grammatical error correction](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3401–3414, Abu Dhabi, UAE. 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.
- Agnes Luhtaru, Taido Purason, Martin Vainikko, Maksym Del, and Mark Fishel. 2024. [To err is human, but llamas can learn it too](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12466–12481, Miami, Florida, USA. Association for Computational Linguistics.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. [Encode, tag, realize: High-precision text editing](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.
- Arianna Masciolini and Elena Volodina. 2025. MultiGEC-2025: Multilingual grammatical error correction shared task. <https://spraakbanken.gu.se/en/compsla/multigec-2025>. NLP4CALL Workshop, NoDaLiDa 2025.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskyi. 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.
- Frank Palma Gomez and Alla Rozovskaya. 2025. [Low-resource grammatical error correction: Selective data augmentation with round-trip machine translation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25749–25770, Vienna, Austria. Association for Computational Linguistics.
- Taehee Park, Heejin Do, and Gary Lee. 2025. [Leveraging what’s overfixed: Post-correction via LLM grammatical error overcorrection](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28195–28207, Suzhou, China. Association for Computational Linguistics.
- Kevin Parnow, Zuchao Li, and Hai Zhao. 2021. [Grammatical error correction as GAN-like sequence labeling](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3284–3290, Online. Association for Computational Linguistics.
- Tom Potter and Zheng Yuan. 2024. [LLM-based code-switched text generation for grammatical error correction](#). In *Proceedings of the 2024 Conference on*

- Empirical Methods in Natural Language Processing*, pages 16957–16965, Miami, Florida, USA. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Vipul Raheja and Dimitris Alikaniotis. 2020. [Adversarial Grammatical Error Correction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3075–3087, Online. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.
- 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.
- Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. 2025. [Adapting LLMs for minimal-edit grammatical error correction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 118–128, Vienna, Austria. Association for Computational Linguistics.
- Xin Sun, Tao Ge, Shuming Ma, Jingjing Li, Furu Wei, and Houfeng Wang. 2022. A unified strategy for multilingual grammatical error correction with pre-trained cross-lingual language model. *arXiv preprint arXiv:2201.10707*.
- Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. [Grammatical error correction using pseudo learner corpus considering learner’s error tendency](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online. Association for Computational Linguistics.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and finetuning](#). *Preprint*, arXiv:2008.00401.
- Maksym Tarnavskyy, 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.
- Yixuan Wang, Baoxin Wang, Yijun Liu, Qingfu Zhu, Dayong Wu, and Wanxiang Che. 2024. [Improving grammatical error correction via contextual data augmentation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10898–10910, Bangkok, Thailand. Association for Computational Linguistics.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for English](#). *Transactions of the Association for Computational Linguistics*, 8:377–392.
- 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.
- Ikumi Yamashita, Satoru Katsumata, Masahiro Kaneko, Aizhan Imankulova, and Mamoru Komachi. 2020. [Cross-lingual transfer learning for grammatical error correction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4704–4715, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Haihui Yang and Xiaojun Quan. 2024. [Alirector: Alignment-enhanced Chinese grammatical error corrector](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2531–2546, Bangkok, Thailand. Association for Computational Linguistics.
- Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. [Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.
- Elaine Zosa, Jouni Louma, Kai Hakala, Antti Virtanen, Mika Koistinen, Risto Luukkainen, Akseli Reunamo, Sampo Pyysalo, and Jonathan Burdge. 2025. Poro 2: Continued pretraining for language acquisition. *LumiOpen*.

A Training Details

Tables 8–10 report the hyperparameters used for each model across both training stages.

Hyperparameter	Stage 1	Stage 2
Optimizer	Adafactor	Adafactor
Learning rate	5×10^{-5}	1×10^{-5}
Weight decay	0.01	0.01
Batch size	8	8
Grad. accum.	4 (eff. 32)	4 (eff. 32)
Max length	512	512
Warmup steps	500	50
Max epochs	2	5

Table 8: FinT5 hyperparameters.

Hyperparameter	Stage 1	Stage 2
Optimizer	AdamW	AdamW
Learning rate	3×10^{-5}	1×10^{-5}
Weight decay	0.01	0.01
Batch size	8	8
Grad. accum.	4 (eff. 32)	4 (eff. 32)
Max length	512	512
Warmup steps	500	50
Max epochs	2	5

Table 9: mBART hyperparameters.

B Zero-shot prompt

The following text is our prompt for LLM:

You are a Finnish grammatical error correction (GEC) system. Your task is to correct grammatical errors in Finnish sentences while making minimal changes.

Rules:

- Fix ONLY grammatical errors (morphology, case endings, verb conjugation, agreement).
- Do NOT paraphrase or change the meaning.
- Do NOT change vocabulary unless the word itself is wrong.
- Make the MINIMAL number of changes necessary.
- Output ONLY the corrected sentence - no explanation, no commentary.
- If the sentence has no errors, output it unchanged.

Hyperparameter	Stage 1	Stage 2
Quantization	4-bit	
LoRA rank	64	
LoRA alpha	128	
LoRA target	All Linear	
LoRA Dropout	0.05	
Num. Base Params	8B	
Num. Trainable Params	167M (2.01%)	
Optimizer	AdamW	
Learning rate	2e-4	1e-5
Max epochs	2	

Table 10: Poro2 hyperparameters. *All linear* refers to gate_proj, o_proj, v_proj, q_proj, up_proj, down_proj and k_proj.

From Metrics to Meaning: Rule-Grounded LLM Explanations for Data Literacy in the Case of Youth Football

Tomasz Piłka and Tomasz Kuczyński and Mateusz Czajka

Adam Mickiewicz University, Poznań

Center for Artificial Intelligence

Faculty of Mathematics and Computer Science

Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland

Abstract

Young athletes, parents, and coaches are increasingly exposed to training metrics from wearable technology, yet such metrics are difficult to interpret without contextual explanation. We present a rule-grounded data-to-text framework for supporting data literacy in youth football through concise, stakeholder-specific summaries of training sessions. A rule layer maps duration-normalised indicators to structured facts about session profile, internal intensity, speed exposure, and movement dynamics, which are then verbalised by a large language model for coaches, parents, or players. We compare direct generation from raw metrics, generation from rule-derived facts, and an augmented rule-grounded configuration, ENRICHED, that supplements validated facts with raw metrics and explicit threshold definitions. In this setting, selected open-weight models are additionally adapted using LoRA. The framework is developed using 122 anonymised player-session records from a U15 environment and evaluated on a held-out subset of ten sessions with stakeholder-oriented reference summaries. The results indicate that rule grounding improves reliability and audience adaptation compared with direct generation from raw metrics, particularly by reducing unsupported or overly strong interpretations. A school-based expert evaluation with physical education teachers further suggests that player-facing explanations in the evaluated ENRICHED setting can remain accurate, comprehensible, and practically useful. We position the framework as an interpretable data-literacy support interface for youth sport analytics.

1 Introduction

Wearable sensors and training analytics platforms are becoming increasingly common in youth sports. As a result, coaches, parents, and young athletes are exposed to growing amounts of numerical information about training load, movement dynamics, and

physiological effort. However, such metrics require context to become understandable and practically meaningful. Stakeholders often encounter dashboards and indicators without sufficient guidance on how to interpret these measurements in relation to a specific training session. This creates a data literacy challenge: translating quantitative signals into concise, context-sensitive explanations.

This challenge is particularly evident in youth football, where the same session may need to be explained differently to different audiences. Coaches require concise technical interpretations that remain faithful to the data, whereas parents and players need clearer explanations that avoid specialist terminology and communicate the practical meaning of the metrics. In this setting, explanation functions as an educational interface that helps stakeholders read and reason about performance data.

This perspective aligns with recent educational Natural Language Processing (NLP) research on stakeholder needs, pedagogical alignment, and human-centred evaluation in Large Language Model (LLM)-based educational systems (Pal Chowdhury et al., 2025a; Galletti and Cesaroni, 2025). Recent work also suggests that pedagogical generation benefits from explicit control over communicative intent, while pedagogical quality remains difficult to assess using automatic methods alone (Petukhova and Kochmar, 2025; Kochmar et al., 2025). These concerns are directly relevant in our setting: when an LLM interprets raw training metrics without additional structure, it may produce explanations that are fluent but unsupported, overly strong, or inappropriate for the intended audience.

To address this issue, we propose a rule-grounded data-to-text framework for supporting data literacy in youth football. A rule layer converts normalised session metrics into interpretable facts about session profile, effort characteristics, and movement dynamics. An LLM then verbalises

these facts for a selected audience—coach, parent, or player—using audience-specific framing and terminology. We compare three generation settings: direct generation from raw metrics, strictly rule-grounded generation, and an augmented rule-grounded configuration, ENRICHED, that supplements validated rule-derived facts with raw metrics and explicit band definitions. In the open-weight experiments, LoRA adaptation is applied only to the selected models in ENRICHED; we therefore treat this condition as an applied configuration rather than as an isolated ablation of the added metric and threshold information.

The framework is developed using 122 anonymised player-session records from a U15 football environment. The controlled evaluation is conducted on a held-out subset of ten sessions for which aligned coach-, parent-, and player-facing reference summaries were prepared. The study addresses three questions: whether explicit rule grounding improves interpretive reliability compared with direct generation from raw metrics; whether the same rule-derived session interpretation can support multiple stakeholder groups while preserving a common session meaning; and whether ENRICHED can provide player-facing explanations that remain educationally useful and practically understandable. Our main contribution is to operationalise automated session summarisation as a stakeholder-oriented data literacy support problem rather than as a generic data-to-text task. By combining domain-informed rule grounding with persona-specific LLM verbalisation, the framework connects wearable-sensor metrics with explanations that are faithful to the data, adapted to the audience, and meaningful in practice.

2 Related Work

Recent work in educational NLP emphasises that LLM-based systems should be designed and evaluated with respect to stakeholder needs, pedagogical goals, and authentic educational workflows, rather than fluency alone (Pal Chowdhury et al., 2025a; Galletti and Cesaroni, 2025; Pal Chowdhury et al., 2025b). This view is consistent with research on data literacy, which shows that indicators and dashboards require contextualised support to become meaningful for users (Konold et al., 2015; Mandinach and Gummer, 2016; Michos et al., 2023; Valle et al., 2021). In youth football, this implies that training metrics should be communicated in

ways that are understandable and appropriate for coaches, parents, and players.

From a generation perspective, our work builds on data-to-text conversion and controlled generation from structured inputs. Classical NLG research has emphasised content determination, content selection, planning, and surface realisation when generating text from data (Reiter and Dale, 2000; Gatt and Krahmer, 2018). These challenges remain central in neural and LLM-based generation, particularly where factual accuracy and numerical consistency are required (Wiseman et al., 2017; Puduppully et al., 2019; Chen et al., 2020; Maynez et al., 2020). Our framework follows this tradition by introducing an explicit interpretive layer between numerical session metrics and natural-language realisation.

Related work also highlights the importance of controlling communicative intent, audience, and readability in generated text (Petukhova and Kochmar, 2025; Moorjani et al., 2022; Tran et al., 2025). This is especially relevant in youth football, where movement and load indicators vary across age, role, and match context (Harley et al., 2010; Atan et al., 2016; Algroy et al., 2021; Palucci Vieira et al., 2019; Dalen and Loras, 2019; Hannon et al., 2021). We extend these perspectives by treating session summarisation as stakeholder-oriented explanatory generation for data literacy support. Rule grounding makes the interpretation traceable and auditable, while LLM realisation adapts the same underlying session meaning to coaches, parents, and players.

3 Rule-Grounded Formulation

Let $x_{i,t}$ denote a player-session record for player i in session t , where t indexes a training or match session rather than continuous time. Each record is represented as a vector of wearable-derived numerical indicators, including measures of external load, internal intensity, speed exposure, and movement dynamics. The goal is to generate a short natural-language explanation y for a target stakeholder $p \in \{\text{coach, parent, player}\}$.

We consider three generation settings that differ in the amount of interpretive structure provided to the language model. In all cases, a feature mapping ϕ preprocesses the raw player-session record and applies duration normalisation to selected indicators, yielding $\phi(x_{i,t})$. A banding function B then maps selected normalised indicators to ordinal lev-

els such as low, medium, high, and elite, using metric-specific threshold definitions Θ :

$$z_{i,t} = B(\phi(x_{i,t}); \Theta) \quad (1)$$

The banded representation $z_{i,t}$ is passed to a rule base R , which produces a set of structured explanatory facts:

$$F_{i,t} = R(z_{i,t}) = \{(f_k, s_k, \pi_k)\}_{k=1}^K \quad (2)$$

Here, f_k is an interpreted claim about the session, s_k is the supporting evidence associated with that claim, such as the relevant metric value and band, and π_k is a priority value used for content selection when multiple facts are available.

Each fact is represented as a compact structured object containing a fact type, supporting metric, interpreted value, and priority level. The rule base encodes domain knowledge about session characteristics, including overall load profile, internal intensity, speed exposure, and movement dynamics. For example, HighDist is activated when distance per minute is assigned to a high band and emits the following load-profile fact: “Distance per minute is high, so this session involved a large amount of running in volume terms. This may support endurance development and sustained lower-limb work, even if there were not many short bursts.” The example shows how the rule layer converts a metric band into an interpretable statement that can later be verbalised differently for coaches, parents, or players. When multiple rules are activated simultaneously, priority-based selection resolves overlap and determines which facts are passed to the generation stage.

The three generation settings differ in the conditioning input provided to the language model:

$$C_m(x_{i,t}) = \begin{cases} \phi(x_{i,t}), & m = \text{RAW}, \\ F_{i,t}, & m = \text{RULES}, \\ (F_{i,t}, \phi(x_{i,t}), \Theta), & m = \text{ENRICHED} \end{cases} \quad (3)$$

The final explanation is produced by a language generation module conditioned on the selected input and the target persona:

$$y = G(C_m(x_{i,t}), p) \quad (4)$$

The generation setting is indexed by $m \in \{\text{RAW}, \text{RULES}, \text{ENRICHED}\}$.

This formulation separates analytical interpretation from linguistic realisation. In the RAW setting,

the language model receives normalised metrics and must infer their meaning directly. In the RULES setting, the rule layer determines the structured interpretation to be verbalised for a given audience. In ENRICHED, rule-derived facts are supplemented with the underlying normalised metrics and threshold definitions. In the implementation evaluated here, the selected open-weight models in this setting are additionally adapted using LoRA.

4 System Overview

4.1 Architecture

Figure 1 shows the proposed pipeline for generating short, stakeholder-oriented explanations from structured training-session data. The system preprocesses wearable-derived metrics into normalised features, maps them to ordinal interpretive bands, and passes them to a rule layer that produces structured explanatory facts about the session profile. A persona-conditioned LLM then generates audience-specific summaries under three input settings: direct generation from normalised raw metrics (RAW), generation from validated rule-derived facts only (RULES), and generation from rule-derived facts supplemented with raw metrics and explicit band definitions (ENRICHED).

In the implementation evaluated here, the selected open-weight models used in the ENRICHED setting are additionally adapted using LoRA. The architecture supports both the RAW–RULES comparison and the applied evaluation of ENRICHED.

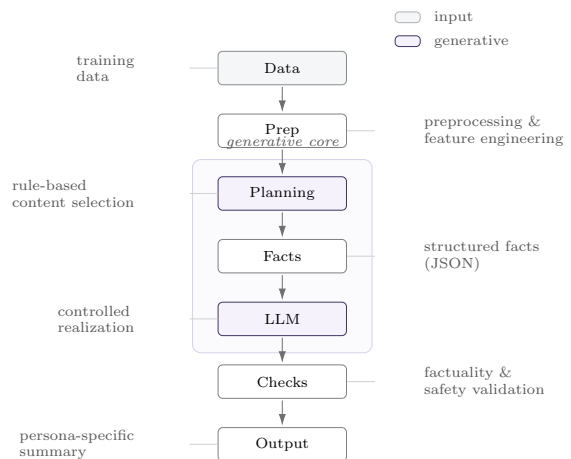


Figure 1: Overview of the proposed persona-specific summary generation pipeline. The central flow represents the main processing stages, while side labels indicate the functional role of each stage. Violet highlights the generative core.

4.2 Rule Layer

The rule layer transforms normalised session features into structured explanatory facts and performs content selection before generation. Selected metrics are mapped to ordinal bands (*low*, *medium*, *high*, *elite*) using metric-specific threshold definitions. These bands provide a traceable intermediate representation for deterministic reasoning about the session profile.

The rule base applies domain-informed conditions to derive compact facts about overall load, internal intensity, speed exposure, and movement dynamics. Each activated rule is represented by a rule identifier, a short explanatory fact, the metric context from which it was triggered, and a priority value used to resolve overlap between simultaneously activated rules. For example, a high distance-per-minute band can activate a load-profile fact indicating substantial running volume. Priority-based selection determines which facts are passed to the generation stage.

In the RULES setting, only the selected rule-derived facts are passed to the generator. In the ENRICHED setting, these facts are accompanied by the normalised raw metrics and the explicit threshold definitions used to derive the bands.

4.3 LLM Realisation

The language model verbalises the selected session interpretation for the target audience under persona-specific style constraints. Generation is prompt-based in all settings: the model receives a structured input for the active condition and an instruction specifying the target persona, expected level of technical detail, and output length. Inputs were serialised as compact JSON-style records containing either the eight metric values, the selected rule-derived facts, or the selected facts together with metric values and threshold definitions.

For coach-facing summaries, the realisation emphasises concise technical interpretation and metric-aware phrasing. For parents, the model uses accessible language and practical framing while avoiding unnecessary technical detail. For players, the summaries use simpler, experience-oriented language adapted to the athlete's perspective.

Across all settings, the LLM expresses the session meaning in audience-appropriate language. The settings differ in how much of this meaning is provided explicitly: in RAW, the model must infer the interpretation from normalised metrics; in

RULES, it receives selected rule-derived facts; and in ENRICHED, these facts are supplemented with supporting numerical metrics and threshold definitions. In the ENRICHED experiments reported here, the selected open-weight realisation models are additionally adapted using LoRA.

5 Data and Preprocessing

The dataset consists of 122 anonymised player-session records collected from 24 U15 players aged 14–15 in a youth football environment. Each record corresponds to one player observed in one training or match session and contains session metadata together with wearable-derived indicators of external load, internal intensity, speed exposure, and movement dynamics. The data were processed at the player-session level, so each instance represents one athlete's session profile rather than an aggregate team-level session.

The analytical vector used by the rule layer contained eight core metrics: distance per minute, high-metabolic-load distance per minute, high-speed-running distance per minute, maximal velocity, accelerations per minute, decelerations per minute, time in heart-rate zones 4–5, and metabolic power per minute. Session duration was retained as a separate metadata field, while selected exposure variables were expressed per minute to support comparison across sessions of different length.

For rule-based interpretation, selected normalised indicators were mapped to ordinal bands (*low*, *medium*, *high*, *elite*). Higher-intensity thresholds were based on youth-football reference values, while lower bands were calibrated using the empirical distribution of the collected data and coach input. The resulting banded representation was used to derive structured explanatory facts.

The processed dataset was split according to the experimental design. A held-out subset of ten player-session records was used for the controlled RAW–RULES comparison and for the evaluation of the ENRICHED setting. The remaining 112 records were used as adaptation data for the LoRA-based implementation of ENRICHED. In this setting, the generator also received normalised raw metrics and corresponding threshold definitions together with the rule-derived facts. Records with incomplete or inconsistent data were excluded from subsequent rule-based analysis. All data were processed in anonymised form.

6 Experimental Design

Rather than evaluating the framework as a broad model benchmark, we assess it as a stakeholder-oriented educational communication system. The evaluation has two complementary stages. First, we compare RAW and RULES generation for coach-facing summaries, where unsupported claims and scale-related errors are particularly salient because the summaries require technical interpretation. Second, we evaluate ENRICHED for player-facing summaries through a school-based expert study. Because these stages address different parts of the framework and use different evaluation procedures, their results are interpreted as complementary rather than directly interchangeable.

The study addresses three questions: whether explicit rule grounding improves reliability compared with direct generation from raw metrics; whether the same rule-derived session interpretation can support multiple stakeholder groups; and whether ENRICHED can produce player-facing explanations that remain educationally useful in the evaluated setting.

6.1 Compared Conditions

We compare three generation conditions.

RAW. In the baseline condition, duration-normalised numerical session indicators are provided directly to the language model, which must infer their meaning during generation.

RULES. In the strictly rule-grounded condition, the rule layer converts the normalised metrics into structured explanatory facts. The language model then verbalises only these selected facts for the target audience.

ENRICHED. In this augmented rule-grounded condition, the model receives the selected rule-derived facts together with normalised raw metrics and the explicit threshold definitions used to assign ordinal bands (*low*, *medium*, *high*, *elite*). The additional information therefore refers specifically to supporting metric values and threshold definitions, rather than to broader contextual variables such as individual condition or training history. In the implementation studied here, the selected open-weight models are additionally adapted using LoRA. This condition therefore combines richer input information, threshold exposure, and model adaptation, and is not treated as an ablation of any single factor.

In all settings, the output is a short natural-

language session summary intended as an explanatory note rather than a full analytical report. Aligned coach–parent–player reference triplets are used as a qualitative anchor for stakeholder alignment rather than as a separate quantitative benchmark.

6.2 Evaluation Data and Reference Set

The full dataset comprises 122 player-session records collected in a U15 football setting. For the controlled comparison, we used a held-out subset of ten sessions with manually prepared, stakeholder-oriented reference summaries for coaches, parents, and players. The selected sessions are as follows: 3, 9, 10, 12, 14, 23, 24, 25, 50, and 89.

The three reference sets provide the same underlying session interpretation in audience-specific formats. Coach summaries explicitly mention performance indicators and their interpretation, parent summaries retain the main message while simplifying terminology, and player summaries use second-person, experience-oriented language. This aligned triplet design supports qualitative assessment of whether generated summaries preserve the same semantic core while adapting to different stakeholder needs.

The same held-out ten-session subset was used for the RAW–RULES comparison and for the evaluation of ENRICHED. The remaining 112 player-session records were used as adaptation data for the LoRA-based implementation described in Section 6.3. This allocation reflects the constraints of the available dataset and keeps the evaluation subset separate from adaptation. Given the size of the held-out subset, the quantitative scores are used primarily as descriptive indicators within this evaluation setting.

6.3 Models and Generation Setup

The study follows a staged evaluation setup. In the first stage, we replicate the RAW–RULES comparison on the ten-session evaluation subset using LLM-as-a-judge and qualitative expert review. This stage is used to assess the effect of explicit rule grounding and to identify models suitable for further adaptation.

On this basis, we selected two models for the ENRICHED stage: **Mistral-3B-Instruct** and **Llama-8B-Instruct**. Mistral-3B-Instruct was selected as the stronger variant within the Mistral family in the replicated comparison, while Llama-8B-Instruct

was retained as a representative of a different open-weight model family. Hosted commercial models were not included in this stage because LoRA adaptation was only feasible for open-weight models in our implementation.

Generation parameters were held constant within each model across the RAW and RULES conditions. All outputs were generated from the same ten-session subset using matched prompts and identical output-length constraints. For the ENRICHED condition, the selected models were adapted with LoRA on the remaining 112 sessions and evaluated on the held-out ten-session subset.

For LoRA adaptation, we constructed persona-specific prompt-completion pairs from the non-evaluation records only. Each input used the same ENRICHED representation as at inference time: a standardised session header, normalised raw metrics, rule-derived observations, threshold definitions, and the persona instruction. Each target completion was a short Polish summary for the selected stakeholder. In our implementation, these target summaries were obtained through teacher-model distillation using the strongest prompt-based system available in the pipeline. The authors then screened them for formatting consistency and obvious factual issues. The held-out evaluation subset was excluded from this adaptation process.

Given the limited adaptation set and the absence of a dedicated development split, we used one conservative LoRA configuration for both adapted open-weight models rather than performing a broad hyperparameter search. We used QLoRA with 4-bit NF4 quantization and bf16 compute, LoRA rank $r = 8$, $\alpha = 16$, dropout = 0.05, and attention-projection target modules (q_proj, k_proj, v_proj, o_proj). Training used paged AdamW 8-bit, learning rate 1×10^{-4} , per-device batch size 1, gradient accumulation 8, 5 epochs, maximum sequence length 2048, and random seed 42.

Generation remained prompt-based in all conditions. Within each summary-generating model, the decoding setup was kept fixed across the compared prompting conditions. For the selected open-weight models carried forward to the ENRICHED stage, we used low-temperature decoding with fixed output-length limits: temperature 0.05 for Mistral-3B-Instruct and 0.07 for Llama-8B-Instruct, with a maximum of 400 generated tokens. Persona prompts and output-length constraints were kept unchanged within each model

family.

Prompt format example. In the ENRICHED setting, the model input was serialised as a compact JSON-style structure containing: (i) a standardised session header, (ii) normalised raw metrics, (iii) rule-derived textual observations, and (iv) threshold definitions. This structured input was followed by a persona-specific instruction controlling audience, expected technicality, and target length.

6.4 Evaluation Dimensions

The evaluation combines quantitative and qualitative perspectives to capture interpretive reliability and educational usefulness across the studied conditions.

Interpretive reliability. First, we assess whether the generated summaries remain faithful to the source session profile. This dimension focuses on factual faithfulness and numerical consistency, with special attention to unsupported inferences, incorrect magnitude interpretation, and misleading summary framing.

LLM-as-a-judge assessment. Second, in the RAW-RULES comparison, we evaluate coach-facing summaries using a fixed LLM-as-a-judge rubric. GPT-4o mini was used as the judge model for all candidate summaries. Summaries are assessed with respect to faithfulness, coach value, narrative quality, output hygiene, conciseness, and language quality, together with an aggregate *Overall* score. Output hygiene refers to whether the summary avoids malformed structure, irrelevant boilerplate, contradictory statements, excessive repetition, and formatting artefacts. The resulting scores are interpreted descriptively, with emphasis on within-generator differences between RAW and RULES.

Reference-based stakeholder alignment analysis. Third, we use the manually prepared coach-parent-player reference triplets as a qualitative interpretive anchor for examining whether generated summaries preserve the same underlying session meaning while adapting terminology, explicitness, and communicative stance across audiences. This component supports qualitative analysis of stakeholder alignment.

School-based expert evaluation. Finally, ENRICHED is evaluated in a school-based expert study focused on player-facing summaries. Nine

physical education teachers from primary, technical secondary, and general secondary schools evaluated both selected models on all ten held-out sessions using a 0–5 scale. The evaluation was blind with respect to model identity, and raters had access to the source session data. The assessed dimensions were: (i) content correctness, (ii) clarity and understandability, and (iii) practical usefulness. An aggregate *Overall* score is additionally reported as the arithmetic mean of these three ratings.

7 Results

7.1 Reference-Based Stakeholder Alignment

The aligned coach, parent, and player reference summaries show that stakeholder adaptation in this task involves more than stylistic variation. Across the ten shared sessions, the three variants preserve the same underlying interpretation while varying terminology, explicitness, and communicative stance. Coach-facing references name relevant indicators such as distance per minute, metabolic power, time in high heart-rate zones, high-speed running (HSR), high-metabolic-load distance (HMLD), and acceleration/deceleration counts. Parent-facing references express the same session meaning through accessible descriptions of effort and rhythm, while player-facing references use second-person, experience-oriented language linked to how the athlete may have experienced the session. Sessions dominated by continuous running volume, internal load, stop–start movement, low load, or speed exposure are therefore consistently described across audiences without changing the underlying interpretation. For example, internally demanding sessions remain framed as heart-rate-intensive rather than speed-driven, whereas stop-start sessions are consistently described as mechanically demanding across the three audience variants. The reference triplets serve as a qualitative anchor for audience alignment, not merely as stylistic examples.

Coach-facing LLM-as-a-judge evaluation. Table 1 reports the coach-facing comparison between the RAW and RULES conditions across all evaluated summary-generating models. Scores are averaged over the ten-session evaluation subset and are interpreted descriptively. *Faith.* refers to *Faithfulness*, or semantic consistency with the source session profile; *Coach* denotes practical coach value; and *Narr.*, *Hyg.*, *Conc.*, *Lang.*, and *Overall* denote narrative quality, output hygiene, conciseness, language

quality, and the aggregate judge score, respectively.

Table 1: LLM-as-a-judge scores assigned by GPT-4o mini for coach-facing summaries, averaged over the ten-session evaluation subset. Rows denote summary-generating models under the RAW and RULES conditions. Scores are on a 0–10 scale. Higher is better on all dimensions.

Model	Variant	Faith.	Coach	Narr.	Hyg.	Conc.	Lang.	Overall
GPT-5.2	RAW	5.2	6.0	5.3	6.7	6.2	7.8	5.1
GPT-5.2	RULES	7.7	7.1	6.8	9.2	7.5	8.8	7.1
Mistral-3B	RAW	2.6	4.1	4.5	2.8	5.2	6.8	2.8
Mistral-3B	RULES	7.7	7.2	6.9	9.3	7.4	9.0	7.1
Mistral-14B	RAW	3.2	4.4	4.5	3.5	5.4	7.0	3.3
Mistral-14B	RULES	6.7	6.8	6.4	7.9	6.9	8.5	6.4
Bielik-1.5B	RAW	0.5	1.3	1.0	0.7	1.9	3.5	0.7
Bielik-1.5B	RULES	4.0	5.0	4.5	4.6	5.4	6.8	3.8
Bielik-11B	RAW	2.3	3.9	3.1	2.3	4.3	6.3	2.3
Bielik-11B	RULES	5.6	5.8	6.0	6.7	6.5	7.8	5.2
Llama-1B	RAW	1.8	2.4	2.5	2.0	3.7	5.5	2.0
Llama-1B	RULES	4.1	5.1	4.7	5.1	5.1	6.5	4.1
Llama-8B	RAW	1.5	2.3	2.1	1.5	3.8	5.7	1.5
Llama-8B	RULES	6.1	6.2	5.7	8.1	6.4	8.1	5.9

Interpretation of Table 1. Across all evaluated summary-generating models, the RULES condition received higher mean *Overall* scores than RAW, with the largest gains in *Faithfulness*, *Coach Value*, and *Output Hygiene*. Improvements in *Language Quality* were smaller, suggesting that rule grounding primarily improved semantic faithfulness, communicative discipline, and practical usefulness rather than surface fluency alone.

Given the ten-session evaluation subset, Table 1 is best interpreted as a within-generator comparison between RAW and RULES, not as a definitive ranking of generators. The highest *Overall* means in the RULES condition were observed for GPT-5.2 and Mistral-3B. These results motivated the selection of Mistral-3B for the augmented ENRICHED stage, with Llama-8B retained as a representative model from a different open-weight family.

School-based expert evaluation of player-facing summaries. Table 2 summarizes the school-based expert evaluation of player-facing summaries in the augmented ENRICHED setting. Nine physical education teachers rated the outputs on a 0–5 scale for *Content Correctness*, *Clarity*, and *Practical Usefulness*. Inter-rater agreement was estimated using ICC(A,9) and indicated limited agreement across dimensions. For *Content Correctness*, *Clarity*, and *Practical Usefulness*, the ICC(A,9) values were 0.434, 0.158, and 0.497 for Mistral-3B, and 0.000, 0.084, and 0.144 for Llama-8B, respectively.

Table 2: School-based expert evaluation of player-facing summaries in the augmented ENRICHED setting. Ratings are on a 0–5 scale. The *Overall* column reports the mean of Content Corr., Clarity, and Pract. Usef. Higher is better on all dimensions.

Model	Content Corr.	Clarity	Pract. Usef.	Overall
Mistral-3B	4.6	3.9	4.5	4.3
Llama-8B	4.4	3.4	4.5	4.1

Interpretation of Table 2. Both models received high teacher ratings for *Content Correctness* and *Practical Usefulness*, suggesting that the player-facing summaries produced in the evaluated augmented setting were factually controlled and practically meaningful. Mistral-3B scored slightly higher on *Content Correctness*, *Clarity*, and *Overall*, while both models obtained the same *Practical Usefulness* score. The comparatively lower *Clarity* ratings indicate that simplifying session meaning for young athletes remains an important design challenge, consistent with treating player-facing generation as both factual summarisation and educational communication.

7.2 Qualitative Error Analysis

The qualitative analysis revealed three recurring error patterns in RAW generation. First, some summaries overstated conclusions not fully supported by the data, especially when high overall cost was verbalised as sustained maximal intensity. Second, some outputs confused the dominant session dimension, for example by describing a session as speed-oriented when the reference interpretation emphasised internal load or continuous volume. Third, persona adaptation sometimes remained superficial, especially when parent- or player-facing outputs retained overly technical phrasing.

The RULES condition reduced these issues by separating interpretation from realisation. Because the dominant session profile was established before generation, the resulting summaries more consistently preserved audience-invariant meaning while still allowing audience-specific wording. This distinction is important for educational communication: a useful stakeholder explanation should differ in form across coach, parent, and player, but not in its underlying interpretation of the session.

In ENRICHED, raw metrics and explicit band definitions provided a richer explanatory input, but also allowed more phrasing variation than the strictly rule-grounded condition. Because this implementation combined enriched conditioning with LoRA-

based adaptation, the results should be interpreted as feasibility evidence for a more flexible applied setup rather than as an isolated estimate of the contribution of enriched conditioning alone. The school-based evaluation suggests that, in this configuration, player-facing summaries can remain factually controlled and practically useful.

8 Discussion

The results support the central design choice of the paper within the evaluated setting: separating analytical interpretation from linguistic realisation can improve the reliability of stakeholder-facing explanations derived from structured training metrics. When language models generate directly from raw numerical inputs, they may produce fluent summaries that overstate conclusions, confuse the dominant session profile, or adapt style without preserving the intended meaning. By contrast, the rule-grounded pipeline constrains interpretation before generation, making the resulting summaries more stable and easier to control.

This effect is especially relevant in the present educational setting, where the goal is not only to produce natural-sounding text, but to help stakeholders understand what recorded metrics mean in practice. The system therefore functions as an explanatory interface between sensor-based analytics and stakeholder understanding. The aligned coach–parent–player reference set further shows that useful adaptation should change wording, level of detail, and communicative stance while preserving the same semantic core across audiences.

The ENRICHED setting adds an applied perspective. It combines validated rule-derived facts, supporting raw metrics, explicit band definitions, and LoRA-adapted open-weight models; the present study therefore cannot isolate the contribution of any single component. In the evaluated setting, the school-based expert evaluation suggests that ENRICHED can produce player-facing summaries that remain correct, understandable, and practically useful.

More broadly, the study supports a methodological point relevant to educational NLP: systems that explain structured numerical data should not be evaluated solely on fluency or stylistic quality. In this context, faithfulness, audience appropriateness, communicative restraint, and educational usefulness are equally important, especially when explanations are intended for young stakeholders

and must remain understandable without becoming overstated, diagnostic, or misleading.

9 Limitations

Several limitations should be acknowledged. First, the dataset comes from a single youth football environment and contains 122 player-session records from a relatively small group of U15 players. The controlled evaluation used a held-out subset of ten sessions, while the remaining 112 records were used for LoRA adaptation. This split reflects the constraints of the available dataset, but limits robust model assessment; we did not conduct cross-validation or repeated-split evaluation. For the coach-facing LLM-as-a-judge comparison, the small ten-session subset precludes strong statistical claims, so the results are interpreted descriptively.

Second, the system relies on a manually designed rule base. While this improves interpretability and control, it requires domain expertise and may hinder transfer to new settings. Duration-normalised indicators and ordinal bands support comparability across sessions, but may reduce information relevant to practical interpretation, such as absolute session duration or individually calibrated physiological zones.

Third, ENRICHED combines selected rule-derived facts, supporting raw metrics, explicit threshold definitions, and LoRA-adapted open-weight models. The present design therefore cannot isolate whether the observed quality of player-facing summaries is attributable to richer input information, threshold exposure, model adaptation, or their combination.

We did not compare LoRA adaptation against few-shot in-context prompting with the same open-weight models. Given the relatively small size of the adaptation set, such a comparison would be important before making stronger claims about the necessity of parameter-efficient fine-tuning. We therefore interpret the LoRA results as evidence of feasibility for a low-resource-adapted configuration rather than as proof that LoRA is superior to few-shot prompting.

Fourth, the school-based evaluation involved nine physical education teachers and focused only on player-facing summaries. Although this provides an educationally meaningful expert perspective, it is not a large-scale end-user study with players themselves and does not provide equally strong evidence for parent- or coach-facing outputs

in ENRICHED.

Finally, the LLM-as-a-judge evaluation was used as a scalable diagnostic assessment, not as a replacement for human evaluation. Although the same GPT-4o mini judge and fixed rubric were used for all candidate summaries, we did not assess judge reliability through repeated judging runs or calibration against a large independent human-rated benchmark. The RAW-RULES comparison and the school-based expert study should therefore be interpreted as complementary rather than directly comparable stages of evaluation.

10 Conclusion

This paper introduced a rule-grounded framework for generating short, audience-appropriate explanations of youth football training data for coaches, parents, and players. By separating validated interpretation from linguistic realisation, the approach can improve control over what is said while allowing explanations to be adapted to different stakeholders. In the evaluated setting, rule grounding increased reliability relative to direct generation from raw metrics, and the evaluated ENRICHED variant produced player-facing summaries that teachers judged to be correct and practically useful. We therefore position the framework not simply as a generation pipeline but as an interface that supports data literacy in educational communication about sports analytics. Future work should test the approach on larger datasets and evaluate its effects on stakeholder comprehension, trust, and learning.

Ethical and Privacy Considerations

The study used anonymised training session data provided by an external partner. No data that could directly identify individual players were used in the generation or evaluation process. The system was designed to facilitate communication and interpretation, rather than for diagnosis or decision-making.

Acknowledgements

The authors would like to thank JUSTWIN sp. z o.o. for their collaboration and support throughout the research process. The authors remain solely responsible for the interpretation of the results and the conclusions presented in this paper.

References

- Erling Algroy, Halvard Grendstad, Amund Riiser, Tone Nybakken, Atle Hole Saeterbakken, Vidar Andersen, and Hilde Stokvold Gundersen. 2021. [Motion analysis of match play in U14 male soccer players and the influence of position, competitive level, and contextual variables](#). *International Journal of Environmental Research and Public Health*, 18(14):7287.
- Siti A. Atan, Andrew Fokkett, and Ajmol Ali. 2016. [Motion analysis of match play in new zealand U13 to U15 age-group soccer players](#). *Journal of Strength and Conditioning Research*, 30(9):2416–2423.
- Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. [Logicnlg: Logical natural language generation from open-domain tables](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.
- Terje Dalen and Havard Loras. 2019. [Monitoring training and match physical load in junior soccer players: Starters versus substitutes](#). *Sports*, 7(3):70.
- Martina Galletti and Valeria Cesaroni. 2025. [From end-users to co-designers: Lessons from teachers](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 505–516, Vienna, Austria. Association for Computational Linguistics.
- Albert Gatt and Emiel Krahmer. 2018. [Survey of the state of the art in natural language generation](#). *Journal of Artificial Intelligence Research*, 61:65–170.
- Marcus P. Hannon, Nicholas M. Coleman, Lloyd J. F. Parker, John McKeown, Viswanath B. Unnithan, Graeme L. Close, Barry Drust, and James P. Morton. 2021. [Seasonal training and match load and micro-cycle periodization in male premier league academy soccer players](#). *Journal of Sports Sciences*, 39(16):1838–1849.
- J. A. Harley, C. A. Barnes, M. Portas, R. Lovell, S. Barrett, D. Paul, and M. Weston. 2010. [Motion analysis of match-play in elite U12 to U16 age-group soccer players](#). *Journal of Sports Sciences*, 28(13):1391–1397.
- Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, K. V. Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. [Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1011–1033, Vienna, Austria. Association for Computational Linguistics.
- Clifford Konold, Trevor L. Higgins, Stephen J. Russell, and Khalil Khalil. 2015. [Data seen through different lenses](#). *Educational Studies in Mathematics*, 88(3):305–325.
- Ellen B. Mandinach and Edith S. Gummer. 2016. [What does it mean for teachers to be data literate: Laying out the skills, knowledge, and dispositions](#). *Teaching and Teacher Education*, 60:366–376.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Konstantinos Michos, Maria-Luisa Schmitz, and Dominik Petko. 2023. [Teachers’ data literacy for learning analytics: A central predictor for digital data use in upper secondary schools](#). *Education and Information Technologies*, 28:14453–14471.
- Samraj Moorjani, Adit Krishnan, Hari Sundaram, Ewa Maslowska, and Aravind Sankar. 2022. [Audience-centric natural language generation via style infusion](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1919–1932, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Sankalan Pal Chowdhury, Nico Daheim, Ekaterina Kochmar, Jakub Macina, Donya Rooein, Mrinmaya Sachan, and Shashank Sonkar. 2025a. [Large language models for education: Understanding the needs of stakeholders, current capabilities, and the path forward](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1–10, Vienna, Austria. Association for Computational Linguistics.
- Sankalan Pal Chowdhury, Terry Jingchen Zhang, Donya Rooein, Dirk Hovy, Tanja Käser, and Mrinmaya Sachan. 2025b. [Educators’ perceptions of large language models as tutors: Comparing human and AI tutors in a blind text-only setting](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 356–374, Vienna, Austria. Association for Computational Linguistics.
- Luiz Henrique Palucci Vieira, Christopher Carling, Fabio Augusto Barbieri, Rodrigo Aquino, and Paulo Roberto Pereira Santiago. 2019. [Match running performance in young soccer players: A systematic review](#). *Sports Medicine*, 49(6):917–931.
- Kseniia Petukhova and Ekaterina Kochmar. 2025. [Intent matters: Enhancing AI tutoring with fine-grained pedagogical intent annotation](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 860–872, Vienna, Austria. Association for Computational Linguistics.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. [Data-to-text generation with content selection and planning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915.

- Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.
- Hieu Tran, Zonghai Yao, Lingxi Li, and Hong Yu. 2025. [Readctrl: Personalizing text generation with readability-controlled instruction learning](#). In *Proceedings of the Fourth Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2025)*, pages 19–36, Albuquerque, New Mexico, US. Association for Computational Linguistics.
- Natercia Valle, Pavlo Antonenko, Kara Dawson, and Anne Corinne Huggins-Manley. 2021. [Staying on target: A systematic literature review on learner-facing learning analytics dashboards](#). *British Journal of Educational Technology*, 52(4):1724–1748.
- Sam Wiseman, Stuart Shieber, and Alexander M. Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sharing is Caring: Advantages of Sharing a Language Background with Learners as an Annotator of Learner Data in UD

Caroline Grand-Clement

Department of Philosophy, Linguistics
and Theory of Science
University of Gothenburg
gusgrandca@student.gu.se

Arianna Masciolini

Språkbanken Text
Department of Swedish,
Multilingualism, Language Technology
University of Gothenburg
arianna.masciolini@gu.se

Abstract

This paper looks at the impact of annotators sharing a language background with learners when annotating learner data using the Universal Dependencies (UD) framework. We perform a study comparing annotations by two different annotators working on sets of L2 Swedish sentences (learner sentences and target corrections) from the Swedish Learner Language corpus (SweLL) written by learners for whom French is a main writing language. The annotators are both L2 speakers of Swedish but have different knowledge of French: one is a native French speaker and the other has no knowledge of French. We find high annotator agreement, which may indicate a non-significant impact, though we qualitatively observe an advantage in sharing language background.

1 Introduction

The Universal Dependencies (UD) framework has been developed in the past years to create a consistent annotation model across languages which includes information about both individual tokens (POS, morphological features) and the grammatical relations between them (de Marneffe et al., 2021). It is an ongoing project nourished by an active community which has developed and clarified both general and language-specific guidelines. The community has also contributed with treebanks, gathering annotated datasets in different languages; as of November 2025, version 2.17 — the last official release at the time of writing — included 339 treebanks across 186 languages (Zeman et al., 2025).

The use of UD as an annotation standard for learner corpora was proposed by Lee et al. (2017b), who argue that rich morphosyntactic annotation can replace explicit error labeling, allowing research on a broader range on L2 phenomena without prejudicing classic error retrieval. To ensure the latter, Lee et al. recommend representing learner data

as parallel L1-L2¹ treebanks composed of learner sentences on one hand, and target corrections on the other. They argue that using these parallel treebanks rather than traditional error-tagged corpora also improves corpus reuse and interoperability.

On this basis, a number of learner treebanks — some parallel, some not — has been developed, e.g. for learner Chinese (Lee et al., 2017a), English (Berzak et al., 2016; Kyle et al., 2022), Italian (Di Nuovo et al., 2022), Greek (Klironomou et al., 2026), Korean (Sung and Shin, 2024) and Swedish (Masciolini et al., 2025a). As is the case for all UD treebanks, learner texts in these datasets are segmented into sentences, tokenized, lemmatized, POS-tagged and annotated for dependencies. Dependency annotation consists of head-dependent links tagged with a *relation type*, which indicates the syntactic role of the dependent with respects to its head. In addition, some of the above mentioned treebanks provide the morphological analysis of each token and/or additional domain-specific annotations and metadata, such as error labels and learner proficiency levels.

Applying the UD framework to learner productions, however, also comes with some challenges related to the idiosyncrasies of interlanguage. This has prompted each annotation effort to develop its own internal annotation guidelines, which in turn led to some inconsistencies between learner treebanks. These are described by Masciolini et al. (2025b), who advocate for harmonized annotation guidelines that take into account learner errors while remaining faithful to UD principles and promoting cross-language comparability.

The annotators of learner data are not always native speakers of the language, but may be L2 speakers themselves. For example, UD_Swedish-SweLL (Masciolini et al., 2025a), a treebank of

¹In this paper, we will use L1 to refer to someone’s native language, and L2 to refer to their non-native language(s), whether it is their second, or third (and so on) language.

learner Swedish, has been entirely annotated by L2 Swedish speakers, rather than native Swedish annotators. In this study, we continued the annotation of the treebank under the same setup in order to investigate whether having a similar linguistic background between annotator and learners impacts annotation quality.

We devised the following research questions:

- How does the annotator’s native language, as compared to the learners’ main writing language, impact their annotation of learner data and ability to identify causes of errors?
- Can the errors made by learners be identified as being significantly caused by their main writing language?

To answer these questions, we used sentences extracted from the SweLL (Swedish Learner Language) corpus (Volodina et al., 2026). This corpus comprises sentences from adult learners of Swedish with different linguistic background. We focused here on those written by learners who have their highest written proficiency in French.

Two annotators — one whose native language is French and the other who has no prior knowledge of French, and who are both L2 speakers of Swedish — were tasked with annotating sentences of L2 Swedish learners whose main writing language (or one of) is French.

We hypothesize that an annotator whose native language is the same as the learners’ will be able to produce more informed annotations than an annotator who has no knowledge of the learners’ native (or best writing) language.

2 Materials and methods

2.1 The SweLL corpus

The data we used consists of sentences written by French speakers extracted from the SweLL (Swedish Learner Language) corpus (Volodina et al., 2019, 2025, 2026). This dataset includes sentences taken from essays written by adult learners of Swedish as an L2. Each essay is pseudonymized, error annotated and associated with a correction hypothesis, i.e. a normalized version of the essay, minimally edited so that it adheres to the norms of contemporary standard Swedish.² In addition, the corpus comes with information about the language

²See respectively Rudebeck and Sundberg (2021) and Rudebeck et al. (2021) for the complete error annotation and normalization guidelines.

background of each learner, as well as about their proficiency in the target language.

In terms of language background, learner metadata specifies both the native language(s) of learners and their best written language(s) (i.e., the language(s) in which the learner has the best written proficiency). A learner can have one or multiple such languages, which often, but not necessarily, coincide with their native language(s). In the case of French, mismatches between L1 and best writing language are particularly frequent: only 59 sentences are authored by French native speakers, while including all sentences where French is listed among the learner’s writing languages leads to a substantially bigger sample (142 sentence pairs).³ Even so, it should be noticed that French is not one of the dominant languages in SweLL, where the most represented L1 is by far Arabic (117 essays), followed by Kurdish, English, Dari and Persian (34-47 essays). Rather than on the amount of data available, our choice of language was based on the linguistic competence of the annotators.

As for proficiency, each learner is assigned an approximate proficiency level (*Nybörjare* ‘beginner’, *Fortsättning* ‘intermediate’ or *Avancerad* ‘advanced’). Our French subcorpus mostly consists advanced-level productions (89 sentence pairs). The remaining 56 pairs are beginner-level, meaning that intermediate level is not represented at all. In-progress work on the SweLL corpus, however, includes the CEFR labeling of each text, intended to provide further nuance and improve cross-language comparability.

2.2 Annotation

Two annotators participated in this project. Both have English and Swedish as L2s. The native language of one of them (AnF) is French while the other (AnI) has no prior knowledge of that language, their native language being Italian. AnI was more experienced in UD annotation than AnF.

We decided to automatically pre-parse and annotate the data with UDPipe 2 (Straka, 2018) with the 2.15 Talbanken model (the most recent model at the time of the study), then went through the annotations manually. Each annotator worked on their own files, to keep a trace of both annotations even after annotation conflict resolution. As for the rest of the SweLL treebank, manual work mainly

³In the resulting subcorpus, L1s other than French are Albanian, Arabic, Berber languages (not further specified), Kirundi and Somali.

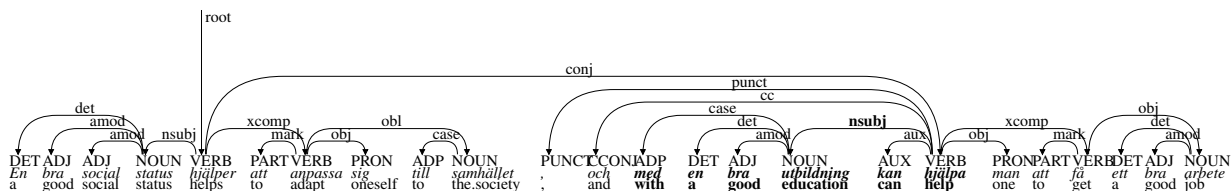


Figure 1: Sentence (1). The noun *utbildning* is annotated as *nsubj* despite the redundant case marker *med*.

revolved around lemmas, Universal POS (UPOS) tags and dependencies, which were all systematically checked and post-edited as needed. Morphological features were only checked in correspondence of learner and parser errors, while language-specific POS tags (XPOS) were ignored and eventually removed from the treebank.

To compare the annotations, meetings were organized to go through an average of 30 sentence pairs per meeting. During these meetings, the two annotators discussed disagreements to resolve them. The equivalent files on the main private SweLL repository were then updated with a final version of agreed-upon annotations. The resulting gold annotations are available as part of UD 2.18.

2.3 Analysis

Differences in annotations of the original learner data were analyzed both quantitatively and qualitatively, while the analysis of the target corrections was only quantitative.

For the quantitative analysis, Inter-Annotator Agreement (IAA) was computed using the Universal Dependencies evaluation script,⁴ comparing the annotations (before conflict resolutions) for the original sentences and those for the target corrections.

We developed a Python script to extract differing line pairs between the annotation files by field. The program allows extraction based on differing lemmas, parts of speech, features, heads or dependency relations. The resulting files were used to bring more detailed information to the quantitative analysis.

During the comparison meetings, notes were made about disagreements on dependencies, lemmas and features that could be due to different interpretations fed by knowledge, or lack thereof, of the learners' writing language. Notes were also taken about sentences upon which the annotators agreed despite there being errors that were possibly due to language interference. These notes served

as the basis for the qualitative analysis.

In the following sections, we assume basic knowledge of the UD annotation framework but link to the official documentation for the particular annotation values that were object of discussion between the annotators.

3 Results

When it comes to original learner sentences, agreement is generally high, the lowest score being morphology-aware labeled attachment score (MLAS, 88.90 aligned accuracy). Most disagreements are about the heads and/or dependency labels, rather than parts of speech (UPOS, 99.02), features (UFeats, 98.42) or lemmas (99.45).⁵

A fifth of the differences in head assignments come from disagreements about punctuation mark heads (25/125). Looking at the disagreements around dependency relations, we can see the most common is one annotator qualifying a relation as *obl*⁶ and the other disagreeing (22/98).

On target sentences, agreement is even higher, with all metrics over 91.00 (cf. also Appendix B). Tokens and words are not 100.00, due to a difference in tokenization of *second hand-butiker*.

The most common disagreement about heads is again about punctuation (19/110), while the most common disagreement about dependency relations is also one annotator qualifying a relation as *obl* and the other disagreeing (18/83).

In sentence (1)⁷, also displayed in Figure 1, there was a disagreement about the relation linking *utbildning* to *hjälp*. During the discussion, AnI argued that the subject may have been omitted, as can be the case in certain romance languages (such as Italian). However, AnF argued that this is less often the case in French, and that *utbildning* may act as a subject, with an unnecessary preposition, following the suggested correction, which was done

⁵The full output of the evaluation script is given in Appendix A.

⁶universaldependencies.org/u/dep/obl

⁷Sentence numbers such as (1) refer to the full learner productions, available in Appendix C along with their corrections.

⁴github.com/UniversalDependencies/tools/blob/master/eval.py

with the full text context, before it was split into individual sentences.

Sentence (2), *Det tar tid för att förklara vad lycka är* [‘It takes time (for) to explain what happiness is’], brought up questions around the relation between *förklara* and the root token *tar*. We reflected that the redundant preposition could have been added as an interference from French, in which a natural translation could be *Il faut du temps pour expliquer* [‘It needs time for explaining’] and agreed on *advcl*⁸ because of the preposition *för*, whereas the analysis of the correction uses the label *csubj*⁹.

While both original and correction are arguably grammatical in sentence (3) (the form of the verb *styr* [‘to rule’] is the same for singular and plural), the use of the word *trenden* [‘the trend’, definite singular] rather than the more idiomatic indefinite plural form *trender* suggested in the correction can be seen as a reflection of the French *la mode* [lit. ‘the fashion’]. However, both AnF and AnI noticed this parallel, as the same construction (*la moda*) also exists in Italian.

There was some discussion about whether to add the *Foreign=Yes*¹⁰ feature to *persons* (4). This token is in place of the Swedish plural form of *person* [‘person’], *personer*. While AnI considered it to be a loan from English, AnF thought it may have been an attempt to create a plural form from the Swedish singular and the French plural formation (which most often adds *-s* at the end of the word). We ultimately agreed on the latter interpretation, thus not adding the *Foreign* feature. This could hint at a need for a more specific feature tag to mark foreign inflection; however, it remains to be seen whether this phenomenon is sufficiently common and well defined to justify a new tag. Similarly, the misspelling *temperature* (instead of *temperatur* (5)) may be a transfer from English, or from the French *température*. Since English is not among the learner’s top-proficiency languages, we annotated it with *Foreign=Yes* and *Lang=fr*¹¹.

Finally, the phrase *som av de* [lit. ‘who/which/that of they’] (6) clearly shows interference from English (‘some of them’). Metadata, where we can see that English is also one of this learner’s writing languages alongside French, also supports this hypothesis. We therefore annotate following the guidelines for English (cf.

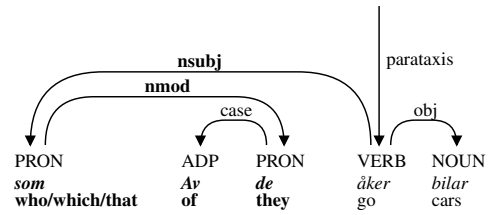


Figure 2: Fragment of sentence (6). The literal translation of the phrase is difficult to analyze, but we argue that *som av de* mimics the English construction ‘some of them’ and annotate accordingly: the relative pronoun *som* is treated as a subject *åker* and the subtree *av de* becomes its nominal modifier.

Figure 2), and including the miscellaneous tag *CalquedLang=en*¹².

4 Discussion

The scores of the annotator disagreement metrics show a strong agreement between the two annotators. It is important to note that the sample under consideration is relatively small (only 139 sentences), which makes generalization difficult.

Common disagreements arose due to AnF’s lesser experience with UD. For example, the disagreements about punctuation heads was often due to AnF’s lack of familiarity with the UD conventions in that regard (i.e., to connect the punctuation to the dependent). Other disagreements of that sort are linked to corner cases using semi-mandatory dependency subtypes (e.g. *nsubj:outer*¹³).

If we take this into consideration, paired with the high agreement observed, this may indicate that the annotators’ different native languages do not significantly impact their annotation as much as their experience in annotating, as relatively few differences are spotted.

From our qualitative analysis, we can see that learner errors due to language interference stem not only from French but also other languages. Sentences (1) to (3) we consider as having interference from French, while sentences (4) and (5) show interferences from a language that could be French or another (such as English). We consider the interference in sentence 6 to have been from English.

Furthermore, there were only few sentences for which these errors led to annotator disagreement. This may show a limited advantage of having a native French annotator as opposed to someone who

⁸universaldependencies.org/u/dep/advcl

⁹universaldependencies.org/u/dep/csubj

¹⁰universaldependencies.org/u/feat/Foreign

¹¹universaldependencies.org/misc.html#lang

¹²The inclusion of this tag in the official UD documentation is currently under discussion; see github.com/UniversalDependencies/docs/issues/1181.

¹³universaldependencies.org/u/dep/nsubj-outer

has no knowledge in French, though we cannot say that this advantage is inexistent: for sentences (1) or (4), knowledge of French grammar and syntax led to a better understanding of the errors.

Language learning and teaching is a complex process of which errors are a necessary part. Our research shows that a learner's L2 Swedish written production can be influenced by their linguistic background as a whole, and not only by their main writing language. This adds to the findings in [Długosz \(2023\)](#), which show that L1 Polish learners of Swedish as a third (or fourth) language produce gender assignment errors which are due to misconstrued gender in their L2 German. In trilingual speakers, non-native languages seem to interfere with each other more than with their native language ([de Bruin et al., 2023](#)).

[Bartolotti and Marian \(2017\)](#) found evidence that vocabulary building when learning a new language depends on each previously known language, which could also explain why some errors (like sentence 5) are difficult to identify as coming from one language or another, especially when the terms in all languages are similar.

In a context where most learners share a common native language (such as native French speakers studying Swedish in France), recognizing such mistakes may lead to the creation of useful teaching tools. However, if there is greater diversity of linguistic backgrounds (such as immigrants studying Swedish in Sweden), spotting errors specific to one's native language may be too ambitious on a class level, though it may still be beneficial at an individual level. Identifying the exact root cause of an error may thus be less important than understanding the linguistic repertoire that a learner already possesses and its possible diversity.

5 Conclusions and future work

In this study we looked at two annotators' annotations of L2 Swedish learner sentences and target corrections to find out if the annotator (AnF) with a similar linguistic background — in terms of being proficient in French writing — as the learners had an advantage over the annotator (AnI) without. The annotation work carried out for this study enriches the UD_Swedish-SweLL with 134¹⁴ new double-annotated sentence pairs, included in the treebank

¹⁴A few of the sentences considered in this study had already been made available as part of the treebank first release. In these cases, the new annotations were compared with the old ones and used to improve the existing dataset.

from the 2.18 UD release (May 2026).

Our analysis of the annotations found that while AnF brought valuable insight for specific cases, the overall disagreement between annotators was small. We also found that it was not simple to identify the language causing errors due to language interference, as learners could and did have non-native languages interfere with their writing production as well. We conclude by highlighting the complexity of the linguistic repertoire learners and the importance of taking the learning context into account.

One could improve on this study by involving more experienced annotators, perhaps only native Swedish speakers and native French speakers. This may limit the disagreements due to differences in UD guidelines interpretation, and narrow down the languages that may impact the interpretation and annotation of learner sentences. Likewise, limiting the sentences studied to those produced only by L1 French speakers may help in identifying how annotators' native languages matching (rather than overlapping) with learners' impacts annotation.

It would be interesting to study specific learner levels as well. At beginner levels, interference from other languages might be greater, and so more important to spot. However, at higher levels, the interference may become more subtle, and thus a better understanding of the learner's language background may be beneficial to spot it.

Finally, reproducing this study with a larger number of sentences, or looking at different native languages, may bring further insight, and help us determine whether there is a significant advantage to matching annotators' native languages to that of the learners' when annotating L2 learner data.

Limitations

The first limitation present in this study is the difference of experience in UD annotation and linguistic analysis between the two annotators, which was the cause of some disagreements due to AnF lack of knowledge of specific dependency subtypes, for example. The number of annotators was also limited, and their native languages (French and Italian) are close linguistically, which may have lessened the influence of their difference. Both annotators were also fluent in other languages aside from Swedish and French (which were the main languages relevant in this study), which may also have had an impact on their annotation.

Another limitation was the fact that many of the

learners, while they all had French as a main writing language, also had other languages either as writing languages or their native language. Only 59/142 sentences were written by L1 French speakers, who possibly have also learned other languages (such as English) before Swedish. However, the SweLL metadata does not allow us to reliably know which additional language(s) the learners know. This may mean that, while French may influence their Swedish production and explain some of the errors, others languages (whether written or spoken) may impact it as well, which is not taken into account in this study’s setup (though discussed in Sections 3 and 4).

Acknowledgments

This work has been supported by Språkbanken — jointly funded by its 10 partner institutions and the Swedish Research Council (2025–2028; project id 2023-00161). We also want to thank Maria Irena Szawerna, one of the annotators of UD_Swedish-SweLL, for her feedback on an earlier version of this paper.

References

- James Bartolotti and Viorica Marian. 2017. [Bilinguals’ existing languages benefit vocabulary learning in a third language](#). *Language Learning*, 67(1):110–140.
- Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. [Universal Dependencies for learner English](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.
- Angela de Bruin, Liv J. Hoversten, and Clara D. Martin. 2023. [Interference between non-native languages during trilingual language production](#). *Journal of Memory and Language*, 128:104386.
- Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. [Universal Dependencies](#). *Computational Linguistics*, 47(2):255–308.
- Elisa Di Nuovo, Manuela Sanguinetti, Alessandro Mazzei, Elisa Corino, and Cristina Bosco. 2022. VALICO-UD: Treebanking an Italian learner corpus in Universal Dependencies. *IJCoL. Italian Journal of Computational Linguistics*, 8(8-1).
- Kamil Długosz. 2023. [The gender incongruity effect in L3 Swedish due to imperfect gender acquisition in L2 German: Are nouns of utter gender more masculine than feminine?](#) *European Journal of Scandinavian Studies*, 53(1):117–137.
- Christina Klironomou, Thelka Pasparaki, Arianna Masciolini, Alexandros Tantos, Despoina Ourania Touriki, Konstantinos Tsiotskas, and Eleni Tsourilla. 2026. Towards Universal Dependencies for L2 learners of modern Greek: Annotation and challenges. In *Proceedings of the Ninth workshop on Universal Dependencies (UDW, LREC 2026)*. Upcoming.
- Kristopher Kyle, Masaki Eguchi, Aaron Miller, and Theodore Sither. 2022. [A dependency treebank of spoken second language English](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 39–45, Seattle, Washington. Association for Computational Linguistics.
- John Lee, Herman Leung, and Keying Li. 2017a. [Towards Universal Dependencies for learner Chinese](#). In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 67–71, Gothenburg, Sweden. Association for Computational Linguistics.
- John Lee, Keying Li, and Herman Leung. 2017b. [L1-L2 parallel dependency treebank as learner corpus](#). In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 44–49. Association for Computational Linguistics.
- Arianna Masciolini, Aleksandrs Berdicevskis, and Maria Irena Szawerna. 2025a. [UD2.17_Swedish-SweLL \[dataset\]](#). Språkbanken Text.
- Arianna Masciolini, Aleksandrs Berdicevskis, Maria Irena Szawerna, and Elena Volodina. 2025b. [Annotating second language in Universal Dependencies: a review of current practices and directions for harmonized guidelines](#). In *Proceedings of the Eighth Workshop on Universal Dependencies (UDW, SyntaxFest 2025)*, pages 153–163. Association for Computational Linguistics.
- Lisa Rudebeck and Gunlög Sundberg. 2021. [SweLL correction annotation guidelines](#).
- Lisa Rudebeck, Gunlög Sundberg, and Mats Wirén. 2021. [SweLL normalization guidelines](#).
- Milan Straka. 2018. [UDPipe 2.0 prototype at CoNLL 2018 UD shared task](#). In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.
- Hakyung Sung and Gyu-Ho Shin. 2024. [Constructing a dependency treebank for second language learners of Korean](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3747–3758, Torino, Italia. ELRA and ICCL.

- Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and 1 others. 2019. [The SweLL language learner corpus: From design to annotation](#). *Northern European Journal of Language Technology*, 6:67–104.
- Elena Volodina, Lena Granstedt, Beáta Megyesi, Julia Prentice, Lisa Rudebeck, Gunlög Sundberg, and Mats Wirén. 2026. [Swell-gold \[dataset\]](#). Språkbanken Text.
- Elena Volodina, Arianna Masciolini, Beáta Megyesi, Julia Prentice, Lisa Rudebeck, Gunlög Sundberg, and Mats Wirén. 2025. [SweLL with pride: How to put a learner corpus to good use](#). *Huminfra handbook: Empowering digital and experimental humanities*.
- Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Jephthé Adolphe, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Arofat Akhundjanova, Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Antonios Anastasopoulos, and 741 others. 2025. [Universal Dependencies 2.17 \[dataset\]](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

A Quantitative annotation: IAA on learner sentences

Metric	Precision	Recall	F1 Score	AlignedAcc
Tokens	100.00	100.00	100.00	
Sentences	100.00	100.00	100.00	
Words	100.00	100.00	100.00	
UPOS	99.02	99.02	99.02	99.02
XPOS	99.66	99.66	99.66	99.66
UFeats	98.42	98.42	98.42	98.42
AllTags	97.61	97.61	97.61	97.61
Lemmas	99.45	99.45	99.45	99.45
UAS	94.67	94.67	94.67	94.67
LAS	92.66	92.66	92.66	92.66
CLAS	91.84	91.97	91.90	91.97
MLAS	88.78	88.90	88.84	88.90
BLEX	91.23	91.35	91.29	91.35

Table 1: Annotator agreement between AnF and AnI’s annotations of the original learner sentences. Sentences org-682-train, org-828-train, and org-1497-train have been excluded as they were split by one annotator and not the other.

B Quantitative annotation: IAA on correction hypotheses

Metric	Precision	Recall	F1 Score	AlignedAcc
Tokens	99.87	99.96	99.91	
Sentences	100.00	100.00	100.00	
Words	99.87	99.96	99.91	
UPOS	99.49	99.57	99.53	99.62
XPOS	99.79	99.87	99.83	99.91
UFeats	98.94	99.02	98.98	99.06
AllTags	98.68	98.77	98.72	98.81
Lemmas	99.74	99.83	99.79	99.87
UAS	95.24	95.32	95.28	95.36
LAS	93.66	93.74	93.70	93.78
CLAS	92.64	92.89	92.76	92.95
MLAS	91.15	91.40	91.27	91.46
BLEX	92.50	92.75	92.63	92.81

Table 2: Annotator agreement between AnF and AnI’s annotations of the corrected target sentences. Sentences trg-682-train, trg-828-train, and trg-1497-train have been excluded as they were split by one annotator and not the other. The tokens and words metrics are not 100.00 due to sentence trg-1210-train, where one of the annotators tokenized ‘second hand-butiker’ into 4 tokens while the other into 2.

C Learner sentences showing language interference and their target corrections

- (1) ORIGINAL
En bra social status hjälper att anpassa sig till samhället , och med en bra utbildning kan hjälpa man att få ett bra arbete .
 A good social status helps to adapt oneself to the.society , and with a good education can help one to get a good job .

CORRECTED
En bra social status hjälper en att anpassa sig till samhället , och en bra utbildning kan hjälpa en att få ett bra arbete .
 A good social status helps one to adapt oneself to the.society , and a good education can help one to get a good job .

‘A good social status helps one to fit into society, and a good education can help one get a good job.’

- (2) ORIGINAL
Det tar tid för att förklara vad lycka är .
 It takes time for to explain what happiness is .

CORRECTED
Det tar tid att förklara vad lycka är .
 It takes time to explain what happiness is .

‘It takes time to explain what happiness is.’

- (3) ORIGINAL
Det är absolut inte trenden som styr hur jag väljer mina kläder utan varför och hur de skulle kunna skydda mig på det bästa sätt .
 It is absolutely not the.trend that rules how I choose my clothes but why and how they would can protect me in the best way .

CORRECTED
Det är absolut inte trender som styr hur jag väljer mina kläder , utan hur de ska kunna skydda mig på bästa sätt .
 It is absolutely not trends that rule how I choose my clothes , but how they will can protect me in best way .

‘It is absolutely not trends that guide how I choose my clothes, but how they will be able to protect me in the best way.’

- (4) ORIGINAL
Sedan ska du boka tid . Därför att de vill bara 25 persons varje dag och det är billigare .
 Then will you book time . Because that they want only 25 persons[misspelled] every day and it is cheaper .

CORRECTED
Sedan ska du boka tid därför att de vill bara ha 25 personer varje dag och det är billigare .
 Then will you book time because that they want only have 25 people every day and it is cheaper .

‘Then you will book a time because they only want to have 25 people every day and it is cheaper.’

- (5) ORIGINAL
De är nödvändiga , särskilt när barnen blir sjuka och deras temperatur stiger på natten och man måste aka till vårdcentralen .
 They are necessary , especially when the.children become sick and their temperature[misspelled] rises at night and one must drive[misspelled] to the.health.center .

CORRECTED

Det är nödvändigt , särskilt när barnen blir sjuka och deras temperatur
It is necessary , especially when the children become sick and their **temperature**
stiger på natten och man måste åka till vårdcentralen .
rises at night and one must drive to the health center .

‘It is necessary, especially when the children gets sick and their temperature rises at night and you need to drive to the health center.’

(6) ORIGINAL

Fölken , kommer till fotbollsplan som Av de åker bilar eller Tar bus
The peoples[misspelled] , come to football.field **that Of them** drive cars or Take bus
eller tog . det är olika Men jag bor precis mitt emot fotbollsplan .
or train[misspelled] . it is different But I live exactly opposite football.field .

CORRECTED

Folk kommer till fotbollsplanen genom att åka bil eller ta buss eller tåg , det
People come to the football.field through at drive car or take bus or train , it
är olika Men jag bor precis mitt emot fotbollsplanen .
is different But I live exactly opposite the football.field .

‘People come to the football field by driving a car or taking a bus or train, it is different but I live right across from the football field.’

Classification of Student Struggle in Mathematics

Hannah Levin*, Madhura Padwal*, and Nchimunya Mwiinga*

Graduate School of Education & Computer Science Department
Stanford University

Correspondence: levinh@cs.stanford.edu

Abstract

Productive struggle is a critical component of mathematics education, requiring students to actively work through ideas rather than just making errors. However, identifying this struggle from text transcripts is challenging because students often mask confusion with epistemic hedging rather than direct statements. Zero-shot large language models exhibit a conservative bias, systematically under-detecting struggle in classroom discourse. We introduce a two-stage NLP pipeline comprising a lexical heuristic gate and an LLM subtype classifier. Our model achieves 90.0% binary accuracy and 84.0% 4-category accuracy. We demonstrate the pedagogical value of this tool by showing that struggle is uniquely concentrated during explicit mathematical reasoning, offering educators a scalable method for root-cause analysis.

1 Introduction

Productive struggle in mathematics involves students actively working through ideas they do not yet fully understand (Hiebert and Grouws, 2007). In classroom discourse, this struggle rarely manifests as explicit statements of inability; instead, students often rely on epistemic hedging, such as "I think" or "maybe", acting as a "shield" against being wrong (Rowland, 1995; Hyland, 1998). Identifying this struggle in real time is particularly challenging because its visibility depends heavily on instructional conditions: struggle surfaces most readily when teachers actively elicit student reasoning, build on student ideas, and create space for back-and-forth discussion. In more teacher-directed lessons, students may not be given the opportunity to verbalize their thinking, leaving the same cognitive roadblocks undetected. Furthermore, even when struggle does surface, its interpretation requires contextual judgment: a delayed

explanation in an introductory lesson signals something very different from the same behavior after repeated modeling and scaffolded practice.

For educators managing large classrooms, conducting root-cause analysis to determine if a student is facing a conceptual misunderstanding or a linguistic barrier is incredibly time-consuming. Scalable Natural Language Processing (NLP) methods offer a promising path forward, enabling systematic analysis of classroom discourse patterns that would be impractical to review manually.

While the NCTE Classroom Transcript Analysis dataset (Demszky and Hill, 2022) provides a rich corpus of elementary math discourse in English, identifying genuine cognitive roadblocks from text alone is challenging. Off-the-shelf NLP models, including zero-shot large language models (LLMs), struggle to accurately parse these highly contextual linguistic markers. As our initial evaluations demonstrated, zero-shot LLMs exhibit a strong conservative bias, systematically under-detecting struggle by defaulting to a "no struggle" classification when faced with ambiguous student language. This bias is further exacerbated by transcription artifacts like "[inaudible]", which models frequently misinterpret as articulatory failure. To address this gap, we introduce a custom, two-stage NLP classification pipeline that identifies and categorizes student struggle into actionable subtypes: Implied Uncertainty, Impasse, and Cannot Explain. By coupling a high-recall, rule-based lexical heuristic gate (Stage 1) with an LLM subtype classifier powered by Gemini (Stage 2), our architecture directly counteracts the conservative bias of baseline models. Evaluated against a rigorously annotated validation dataset, our two-stage pipeline achieves 90.0% binary accuracy and completely eliminates false positives. Furthermore, exploratory application of the model to a holdout set of explicit mathematical reasoning reveals that 37.7% of these turns contain productive struggle. Crucially, our analysis demonstrates

*Equal contribution.

that this struggle is characterized by a stark linguistic shift: struggling utterances exhibit a significant drop in mathematical vocabulary density (averaging 0.55 math words) compared to fluent reasoning (0.87 math words). Ultimately, this model provides a scalable method for educators to conduct root-cause analysis, distinguishing conceptual gaps from linguistic barriers to better support student learning at scale.

2 Prior Literature

Our approach to detecting and classifying student difficulty sits at the intersection of mathematics education research and natural language processing for classroom discourse. We draw upon two primary bodies of work to motivate our construct definition and computational architecture.

In mathematics education, productive struggle is defined as the intellectual effort students expend to make sense of mathematical concepts that are not immediately apparent (Hiebert and Grouws, 2007; Warshauer, 2015). Crucially, this cognitive effort rarely manifests as explicit declarations of inability. Instead, students frequently operate within a "Zone of Conjectural Neutrality" (Rowland, 1995, 2000; Wagner et al., 2015), a cognitive space where they explore ideas they are not yet fully willing to assert. Within this zone, struggle is heavily characterized by epistemic hedging—the use of tentative language such as "maybe," "I think," or "kind of" (Hyland, 1998). Children use these linguistic hedges as a "shield" against the risk of being incorrect (Rowland, 1995).

Another signal of cognitive mathematical struggle is self-correction, where students stop mid-sentence to rephrase or restart. This kind of mid-turn repair often reflects active thinking rather than confusion (Schegloff et al., 1977), and in mathematics, it tends to appear during explanation and justification (Warshauer, 2015).

Pronoun use also matters. In mathematics classrooms, personal pronouns like "I," "we," and "you" serve complex functions related to ownership, authority, and generalization (Pimm, 1987; Rowland, 1999). When students shift from direct language like "I found" to vague phrasing like "you just do this" or "it works," this can reflect uncertainty about whether their approach is correct or generalizable (Rowland, 1999). Conversely, more agentive first-person language may emerge as understanding stabilizes.

Our operationalization of struggle builds directly on this framework, separating explicit breakdowns in articulation (Cannot Explain) or help-seeking (Impasse) from the far more common, but subtle, hedging behaviors (Implied Uncertainty). These categories are described in detail in the Validation Dataset section below, where we present our codebook and annotation framework.

While several educational dialogue datasets exist, none directly address the detection of productive struggle as a cognitive state in whole-class elementary mathematics discourse. The TalkMoves dataset (Suresh et al., 2022) annotates both teacher and student discursive moves across 567 K-12 mathematics transcripts, but its focus on accountable talk practices centers teacher behavior rather than student cognitive states. The Multi-turn Classroom Dialogue (MCD) dataset (Chen et al., 2024) targets mastery prediction from student-tutor dialogue, but is drawn from one-on-one tutoring sessions rather than whole-class instruction, and measures knowledge mastery rather than in-the-moment struggle. Unlike tutoring, where help-seeking is the explicit purpose of the interaction, whole-class discourse requires students to publicly commit to mathematical claims under social risk, which are precisely the conditions that produce the epistemic hedging and articulatory breakdown that our pipeline targets. The NCTE corpus (Demszky and Hill, 2022) is uniquely suited to our task: it captures authentic, whole-class elementary mathematics discourse during cognitively demanding instruction, providing the naturalistic linguistic environment necessary for studying how struggle surfaces in student talk.

Recent advancements in NLP have enabled the large-scale analysis of instructional dynamics. The release of corpora such as the NCTE dataset (Demszky and Hill, 2022) and the TalkMoves dataset (Suresh et al., 2022) has facilitated the automated extraction of teacher dialogue moves and student reasoning patterns. However, detecting nuanced affective or cognitive states, such as uncertainty or struggle, remains an open challenge.

Prior work has demonstrated that identifying student uncertainty in spoken educational dialogue is inherently difficult, often yielding moderate inter-rater reliability even among human experts (Forbes-Riley and Litman, 2011). While recent studies have shown that zero-shot LLMs can approximate human annotation quality on certain text-classification tasks (Gilardi et al., 2023), highly

context-dependent discourse phenomena remain a vulnerability. As our baseline evaluations indicate, when faced with the pragmatic ambiguity of episodic hedging in transcripts stripped of prosodic cues, zero-shot LLMs exhibit a severe conservative bias. Our two-stage classification pipeline builds upon these findings, leveraging a high-recall lexical heuristic to bypass the zero-shot conservative bias while preserving the LLM’s capacity for nuanced subtype classification.

Despite the rich theoretical grounding of productive struggle in mathematics education, the existing literature remains predominantly qualitative and teacher-centered (Young et al., 2023). On the computational side, NLP work on classroom discourse has focused primarily on teacher discourse moves and instructional quality (Suresh et al., 2022; Alic et al., 2022) rather than student cognitive states. Adjacent work on automated confusion detection relies on multimodal signals, prosodic, facial, and acoustic features, that are unavailable in transcript-only corpora (Ma et al., 2024). Text-based approaches to student challenge detection exist in collaborative learning contexts (Suraworachet et al., 2024) but have not been applied to the specific construct of productive struggle in whole-class elementary mathematics. This paper addresses that gap directly.

3 Data

3.1 Dataset

For this study, we used the NCTE Classroom Transcript Analysis dataset (Demszky and Hill, 2022), which is comprised of transcripts for fourth and fifth-grade mathematics instruction in four U.S. school districts serving largely historically marginalized student populations. The transcribed lessons cover core elementary mathematical concepts, including multi-digit multiplication, long division, decimals, and fractions, providing a rich linguistic environment for analyzing student reasoning. Lessons were audio-recorded in classrooms using a capture system that included a lapel microphone worn by the teacher and microphones positioned to pick up student talk. Recordings were fully transcribed by professional human transcribers and anonymized (Demszky and Hill, 2022). Student turns may include multiple speakers and do not track individual students over time. The transcripts capture spoken discourse during whole-class instruction only. They record if the speaker is

a student or a teacher, in what order, and for how long, but do not represent written work, gestures, visual representations, or small-group interactions. As a result, the dataset reflects a partial view of mathematical activity, centered on publicly audible talk in teacher-led discussions.

The original purpose of the dataset was to examine instructional quality and teacher effectiveness, particularly how classroom practices relate to observation scores and student learning outcomes, rather than to study student struggle directly (Kane et al., 2015).

Because struggle is not explicitly labeled in our transcripts, we rely on these structural and lexical features as indirect markers: the timing of reasoning, the length and frequency of student turns, and patterns of tentative or evolving explanation. These features do not measure struggle directly, but prior research suggests they are reasonable signals of students working through uncertainty in whole-class discussion.

3.2 Task Definition: Operationalizing Struggle

Our primary NLP task was the automated detection and classification of student struggle within classroom discourse. We defined struggle as the intellectual effort expended to make sense of mathematical concepts or tasks that are not immediately apparent. To make this construct analytically tractable and pedagogically actionable, we operationalized it into four categories, extracted from the literature:

- **Implied Uncertainty:** Instances where students distance themselves from a mathematical proposition using hedging language (e.g., "maybe," "I think") to protect against potential errors, operating within a "Zone of Conjectural Neutrality" (Rowland, 1995, 2000).
- **Impasse (Call for Help):** Utterances containing clear evidence that a student is blocked from proceeding with a mathematical task, such as explicit statements of confusion or direct requests for help.
- **Cannot Explain:** Moments where a student has an answer but explicitly demonstrates an inability to articulate their reasoning or justify their thinking (e.g., "I just guessed").
- **No Struggle:** Fluent reasoning, procedural questions, or purely logistical discourse.

3.3 Sampling Strategy

To ensure annotators evaluated moments of genuine cognitive demand rather than procedural or off-task talk, we first filtered the corpus using `edu_convokit` to isolate turns containing explicit student mathematical reasoning. However, because struggle is rare even within reasoning turns, purely random sampling would still yield too few instances of minority categories for reliable annotation or model evaluation. We therefore used a hybrid stratified sampling approach across four buckets—one per codebook category—following [Suresh et al. \(2022\)](#), who address the same class imbalance problem when annotating rare discursive moves in K-12 math transcripts. Importantly, bucket assignment determined only which utterances were selected, not how they were labeled; annotators coded all items against the codebook without knowing their source bucket, preserving the independence of ground truth labels from the sampling mechanism. Because struggle markers are highly context-dependent, we extracted 11-turn contextual windows around each target utterance to preserve the surrounding teacher prompts and student responses needed for accurate coding. See Figure A in the appendix for examples of annotated target utterances among their contextual windows.

3.4 Annotation and Reliability

The development of our gold-standard validation set required an iterative codebook revision process to capture the nuanced realities of classroom dialogue. Our final gold-standard validation set consisted of 100 annotated student utterances. This dataset was developed in two distinct phases: an initial set of 50 utterances from our first sampling iteration, and a fresh set of 50 utterances generated using our finalized stratified sampling pipeline.

To ensure robust ground truth labels, all utterances were independently annotated by multiple raters. For the initial 50 utterances, three research team members (the authors) annotated all items independently. To provide critical practitioner validation, we also recruited two practicing elementary school mathematics teachers who split the initial set, annotating 25 items each. For the fresh set of 50 utterances using our finalized stratified sampling pipeline, the three research team members independently annotated all items under the revised codebook. A held-out test set of 50 items was drawn from the same stratified sampling pipeline

and underwent identical independent annotation and consensus review by all three authors. The unannotated holdout ($n=159$) is an entirely separate corpus drawn from the remaining NCTE transcripts after all annotated sets were finalized and was used solely for exploratory analyses reported in Figures 5 and 6.

The development of this dataset required an iterative codebook revision process. Following the initial round of independent annotation, our team conducted consensus review sessions. This collaborative re-annotation yielded several critical clarifications: context-sensitive words such as "could" must be evaluated based on surrounding dialogue rather than automatically treated as hedges; Implied Uncertainty must be grounded in observable hedging language within the utterance itself; and Cannot Explain requires visible articulatory failure.

To address frequent misclassifications of confirmation-seeking questions, we introduced a "cover-up test" in our codebook: if removing the question mark leaves a substantive mathematical claim, the utterance is coded as Implied Uncertainty; otherwise, it is an Impasse. Two items could not be resolved and were excluded, leaving 98 usable validation items. After consensus, human inter-rater reliability on the fresh 50 items reached Krippendorff's $\alpha = 0.961$. See Figure 3 for an overview of the iterative pipeline development process.

4 Methods

4.1 Baseline Evaluations

Before developing our custom architecture, we evaluated four off-the-shelf approaches. A supervised classifier using sentence-level embeddings (all-MiniLM-L6-v2) performed at chance (binary $\kappa = 0.098$), confirming our dataset is too small for standard supervised methods. BART-large-MNLI collapsed to predicting a single category for 92% of utterances ($\kappa = 0.03$). Claude Sonnet 4 and Gemini 3 Pro both fell within the human agreement range ($\kappa = 0.52$ and 0.53) but exhibited a consistent conservative bias, missing genuine struggle 25-30% of the time by defaulting to No Struggle when student language was ambiguous. Critically, the two LLMs agreed with each other at $\kappa = 0.87$, making nearly identical errors—adding a second LLM provided no independent signal. This motivated a two-stage architecture: a lightweight rule-based heuristic handles binary detection cheaply and at

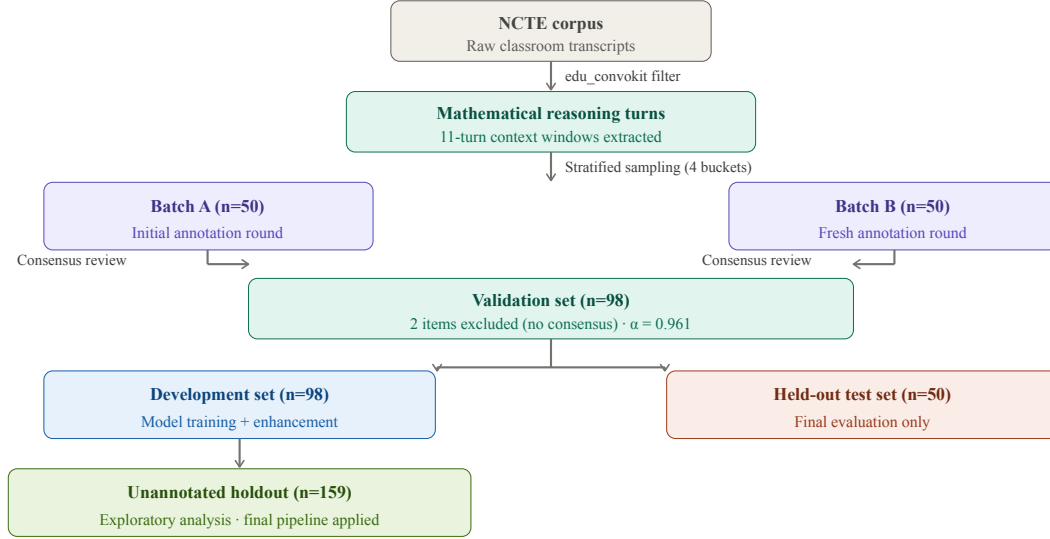


Figure 1: Pipeline for constructing and validating the annotated dataset

high recall, passing only flagged utterances to the LLM for subtype classification. This keeps the system computationally efficient while ensuring the LLM operates on an input where struggle is already the majority class, directly counteracting the conservative bias.

4.2 A Two-Stage Classification Pipeline

To address this conservative bias, we designed a custom, two-stage classification architecture that separates the binary detection task (Struggle vs. No Struggle) from the nuanced sub-categorization task.

Stage 1: High-Recall Lexical Heuristic Gate

The first stage serves as a binary gate optimized strictly for high recall, ensuring that instances of struggle are not permanently lost early in the pipeline. This stage employs a seven-step, rule-based lexical heuristic that assigns weighted scores across three pattern dictionaries derived directly from our codebook. To overcome the limitations of simple keyword matching, we engineered several context-aware components into the heuristic:

- **Confirmation-Seeking Question Detector:** The system applies a "cover-up test" to utterances ending in a question mark. If removing the question mark leaves a substantive mathematical claim, the utterance is flagged as Implied Uncertainty rather than a procedural question.

- **Articulatory Failure Detection:** The heuristic distinguishes genuine self-correction from transcription artifacts by counting textual dashes, specifically subtracting those that co-occur with [inaudible] markers.
- **False Positive Filters:** The system suppresses context-dependent trigger words when used in non-struggle contexts, such as the word "help" used to describe a mathematical strategy (e.g., "I used the number line to help me") or "like" used in geometric comparisons.

Stage 2: LLM Subtype Classifier

Utterances flagged as struggle by Stage 1 are passed to Stage 2 for subtype classification (Implied Uncertainty, Impasse, Cannot Explain, or No Struggle). By pre-filtering the input, Stage 2 evaluates a class distribution where struggle is the majority, directly counteracting the LLM's baseline conservative bias.

This stage utilizes Gemini 3 Pro powered by a highly structured prompt. The prompt embeds all fourteen rules from our revised codebook and includes 50 few-shot examples targeting known boundary cases. It also includes an explicit anti-conservative-bias instruction, directing the model to prioritize finding the struggle over explaining it away.

Finally, our error analysis of early pipeline iterations revealed that LLMs systematically hallucinated articulatory failure when processing brack-

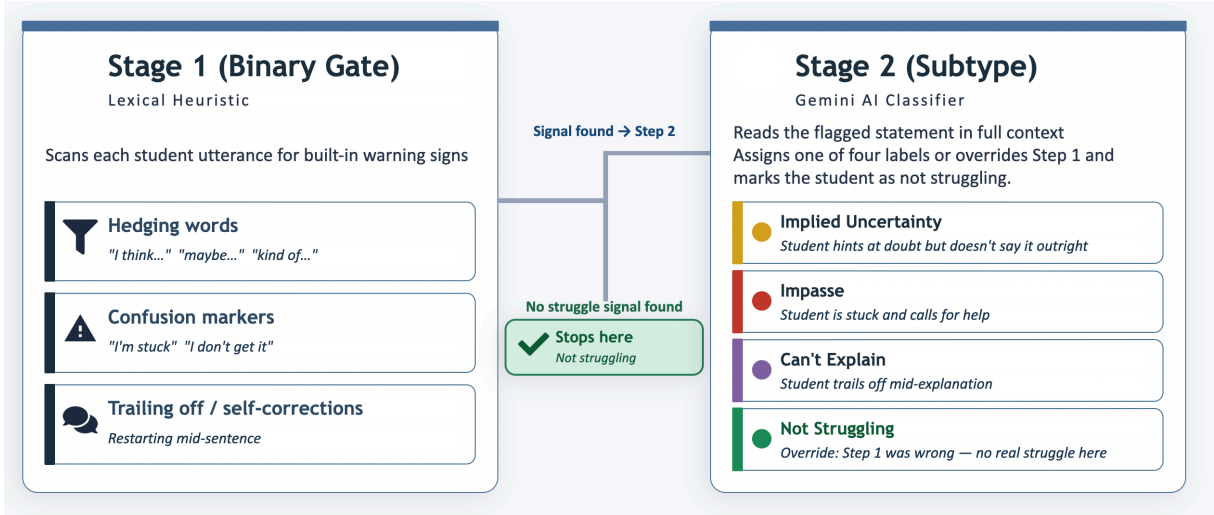


Figure 2: The two-stage struggle classification pipeline. A high-recall lexical heuristic (Stage 1) filters utterances for surface-level warning signs before passing flagged items to a Gemini LLM (Stage 2) for context-aware subtype classification and false-positive correction.

eted transcription markers (e.g., classifying an utterance with an [inaudible] tag as Cannot Explain). To eliminate this noise, all bracketed annotations were programmatically stripped from the context windows before being processed by Stage 2.

5 Results

We report final model performance on the held-out test set ($n = 50$), which was set aside at the start and never used during pipeline development. Table 1 shows how the pipeline improved through development on the validation set, while Table 2 details the per-category performance on the actual test set. The final two-stage pipeline reached an 84.0% four-category accuracy (Cohen’s $\kappa = 0.716$) and a 90.0% binary accuracy in distinguishing struggle from non-struggle. In comparison, the baseline—a zero-shot Gemini model run as a standalone classifier—reached only 68.0% four-category accuracy and 76.0% binary accuracy on the same test set.

Crucially, the two-stage architecture successfully mitigated the LLM’s baseline errors. By offloading the binary decision to the high-recall heuristic, the pipeline completely eliminated false positives; it never incorrectly flagged a non-struggling student as struggling. Total errors on the test set dropped by half, from 16 to 8.

As illustrated by the comparative F1 scores in Table 4, the most significant performance gains occurred in the minority struggle categories. The pipeline improved the F1 score for *Impasse* by 27

points (from 0.40 to 0.67), *Implied Uncertainty* by 24 points (from 0.36 to 0.60), and *Cannot Explain* by 20 points (from 0.58 to 0.78). The *No Struggle* category also saw a 12-point increase, reaching a final F1 of 0.92. Despite these substantial improvements, *Implied Uncertainty* remained the hardest category for the pipeline (F1 = 0.60). Through our error analysis, we found that these categories easily blur together in natural speech; the model plausibly often could not distinguish whether a student was simply expressing uncertainty or entirely lacking the words to articulate the concept.

Finally, we noted that categories such as *Impasse* and *Implied Uncertainty* appeared very rarely in the test set. Because a single misclassification could shift the F1 score by 10 to 30 points, we interpreted these specific metrics as general signs of pipeline improvement rather than absolute measures of the model’s maximum performance.

6 Analysis

The evaluation demonstrated that the final model (84% accuracy, $\kappa = 0.716$) successfully mitigated the conservative bias observed across all off-the-shelf baselines. We found that by offloading binary detection to a high-recall lexical heuristic, Stage 2 received a class distribution where struggle was the majority, which directly counteracted the LLM’s tendency to default to *No Struggle*. Furthermore, the error analysis revealed that the LLMs had conflated [inaudible] transcription tags with genuine articulatory failure, leading them to incorrectly la-

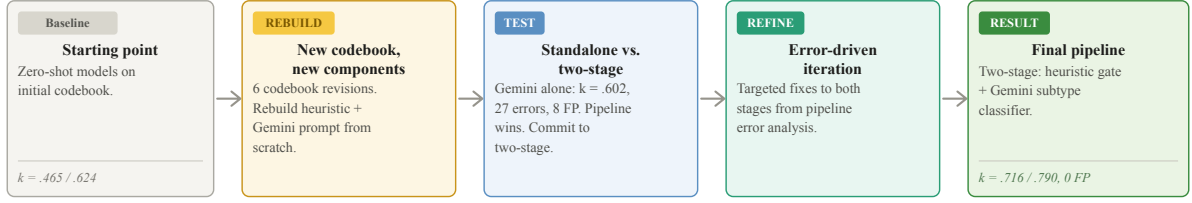


Figure 3: Pipeline development: from codebook revisions to architecture selection and final error-driven refinement

Model / benchmark	4-cat κ	Binary accuracy	Binary κ
Gemini standalone (baseline)	0.602	83.5%	0.657
Two-stage pipeline (Enh. 1)	0.766	95.9%	0.916
+ stripped brackets (Enh. 2)	0.827	95.0%	0.897
Human pre-discussion	0.534*	—	0.607*
Human post-discussion	0.961*	—	—

Table 1: Pipeline development on the validation set (n=98). Human benchmarks are Fleiss’ κ (3 raters). Model kappa is Cohen’s κ against consensus ground truth. These use different evaluation conditions and are not directly comparable.

* Fleiss’ κ , 3 raters, fresh 50 items.

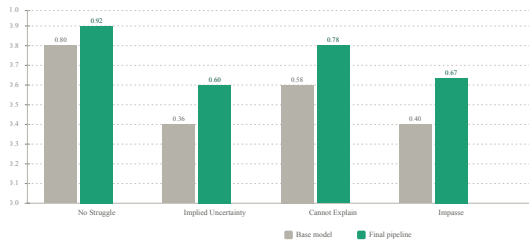


Figure 4: F1 score by category, base model vs. final pipeline (held-out test set, n=50).

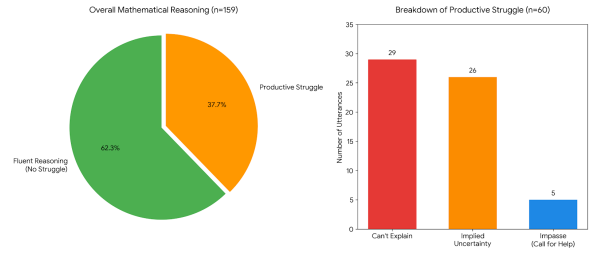


Figure 5: Distribution and categorical breakdown of student mathematical struggle

bel hedged utterances as *Cannot Explain*. It was observed that stripping bracketed annotations prior to Stage 2 resolved this issue and yielded the highest overall accuracy.

Regarding the eight remaining errors, the analysis indicated that seven persisted across both models, pointing to three irreducible limitations of text-only classification. It was noted that three of these errors involved the *Implied Uncertainty* and *Cannot Explain* boundary; both stages confused trailing dashes and restarts because distinguishing them required reading communicative intent, which surface patterns could not resolve. Finally, we observed that one *Cannot Explain* item failed Stage 1 entirely because the breakdown was semantic rather than lexical—the student produced fluent, unhedged speech, but their reasoning was mathematically incoherent. We concluded that further gains would require audio features, cleaner tran-

scription, or the collapsing of *Implied Uncertainty* and *Cannot Explain* into a single category.

To evaluate pedagogical utility at scale, we applied the final model to 159 previously unseen mathematical reasoning turns filtered from the NCTE corpus using edu_convokit. As shown in Figure 5, the model flagged 37.7% (60 of 159) as containing productive struggle – consistent with the pipeline’s 90% binary accuracy and reflecting a corpus already filtered for cognitively demanding turns. *Cannot Explain* was the most prevalent subtype (29 instances, 48%), followed by *Implied Uncertainty* (26 instances, 43%), while *Impasse* was the least frequent (5 instances, 8%). Students rarely voiced explicit calls for help; instead they either attempted an answer they could not fully justify, or hedged tentatively to protect against being wrong.

As seen in Figure 6, fluent reasoning turns av-

Category (n)	Base model			Final pipeline		
	Precision	Recall	F1	Precision	Recall	F1
No Struggle (31)	83%	77%	80%	93%	90%	92%
Implied Uncertainty (6)	40%	33%	36%	75%	50%	60%
Cannot Explain (10)	50%	70%	58%	69%	90%	78%
Impasse (3)	50%	33%	40%	67%	67%	67%
Macro avg	56%	53%	54%	76%	74%	74%

Table 2: Per-category performance on the held-out test set ($n = 50$)

Note: Small support for Implied Uncertainty ($n=6$) and Impasse ($n=3$) means a single item shifting changes F1 by 10-30 points. Read these figures as directional.

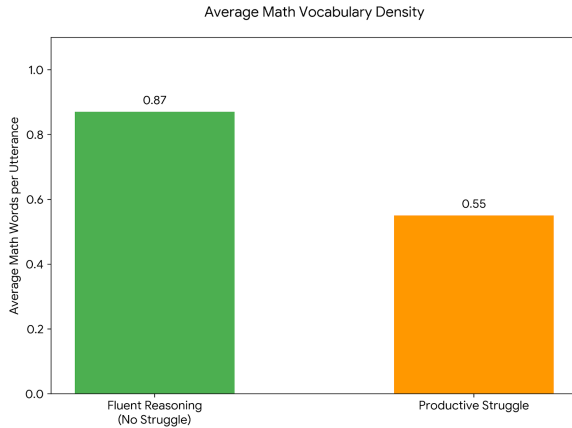


Figure 6: Average mathematical vocabulary density of student utterances using edu_convokit’s get_math_density() function

eraged 0.87 domain-specific math words per utterance, while struggle-flagged turns averaged only 0.55 – a 37% drop. This aligns with the prevalence of Cannot Explain and Implied Uncertainty: when students hit a conceptual wall, they abandon formal math vocabulary and fall back on everyday language or extreme brevity. For educators, this proves the value of the tool for root-cause analysis: by isolating these moments, teachers can discern whether a student requires conceptual reteaching or simply needs linguistic scaffolding and sentence starters.

7 Conclusion

Identifying productive struggle in real-time classroom discourse is a complex challenge, primarily because students frequently mask their cognitive roadblocks with epistemic hedging. As our evaluations demonstrated, zero-shot large language models are ill-equipped to handle this pragmatic ambiguity, exhibiting a conservative bias that systematically under-detects genuine struggle. By

implementing a custom, two-stage classification pipeline that pairs a high-recall lexical heuristic with a precision-tuned LLM, we successfully counteracted this bias. Our final architecture achieved an 84.0% 4-category classification accuracy and entirely eliminated false positives, ensuring the system never incorrectly flags a non-struggling student.

Beyond the technical architecture, our exploratory analysis confirmed the pedagogical value of this approach. The data revealed that explicit mathematical reasoning carries a substantial cognitive load, with productive struggle occurring in 37.7% of observed student participation. Furthermore, this struggle is characterized not just by hesitation, but by a measurable drop in domain-specific mathematical vocabulary, falling from an average of 0.87 to 0.55 math words per utterance. By successfully isolating these moments at scale, our pipeline provides a reliable mechanism for root-cause analysis, empowering educators to distinguish between conceptual misunderstandings and linguistic barriers.

8 Future Work

Because our model moves beyond binary detection and classifies the nature of the struggle (e.g., Implied Uncertainty, Impasse, Can’t Explain), it offers highly targeted insights that can empower educators to make precise pedagogical adjustments.

Rather than functioning as an evaluative measure, this model can be used to power a post-lesson reflective dashboard for teachers. By processing classroom audio or transcripts, the dashboard could provide educators with a high-level summary of discourse trends. For example, it could highlight that "Your students showed high levels of Implied Uncertainty during today’s fraction introduction." This preserves teacher agency; it does not tell the

educator how to teach, but rather cues them to areas requiring curriculum adaptation. If the data shows high rates of Impasse, the teacher may need to revisit foundational concepts. Conversely, if students are displaying high rates of Implied Uncertainty, the teacher might focus on building mathematical confidence and reinforcing a classroom culture where students feel safe asserting their knowledge.

At the same time, the struggle could be a language barrier and not conceptual. If the model identifies a cluster of Can't Explain moments, this signals to the teacher that students might actually grasp the underlying math but lack the academic vocabulary to articulate it. Teachers can use this specific feedback to shift their scaffolding strategy by providing linguistic support, such as sentence starters or targeted vocabulary review, rather than reteaching the mathematical concept itself.

A critical direction for future work is extending this pipeline beyond English-language classrooms. The current model is trained exclusively on English transcripts and relies on lexical markers of epistemic hedging that are English-specific. Different languages encode uncertainty and tentativeness through distinct grammatical and discourse mechanisms. Future work should investigate what text-based markers of productive struggle look like across typologically diverse languages, and whether a universal framework for struggle detection is achievable or whether language-specific codebooks are necessary.

Translating this backend classification model into a real-world, formative feedback dashboard will require dedicated co-design projects with teachers. Engaging directly with educators in this next phase of development will be critical to ensure that the resulting tool respects teacher agency, integrates smoothly into instructional workflows, and provides equitable, actionable insights.

9 Limitations

While our two-stage pipeline demonstrates strong performance, this approach has several methodological and technical limitations. First, our model relies entirely on text-based transcripts, which inherently strips away critical acoustic-prosodic features and non-verbal cues. Prior research has demonstrated that prosodic cues—such as rising intonation, trailing off, and physical hesitation—are significant predictors of student uncertainty in spoken educational dialogue. Because our pipeline

evaluates text alone, it systematically misses instances of implicit struggle conveyed through tone of voice or body language. Second, our evaluation is constrained by a relatively small and homogeneous ground-truth validation set drawn exclusively from U.S. fourth and fifth-grade mathematics classrooms. Consequently, performance metrics for rare categories, such as Cannot Explain (which comprised only 9% of our validation dataset), suffer from base rate instability. Expanding the dataset across different grade levels and instructional contexts is necessary to stabilize the evaluation of these sub-types.

Third, our reliance on a majority-vote human consensus for ground-truth labels introduces an artificial ceiling on model performance. In cases of genuine boundary ambiguity, the model is penalized for agreeing with a minority human rater, even when that rater's interpretation may be pedagogically valid.

Fourth, our model exhibits an asymmetric error profile by design. In optimizing the pipeline to eliminate false positives, the model retains a slight conservative bias, producing a small number of false negatives. In a real-world deployment, this means the system will occasionally miss a struggling student before it ever incorrectly flags a fluent one.

Finally, a core limitation of this evaluation is the small held-out test set ($n=50$) and the very low support for minority categories — Impasse appears only 3 times and Implied Uncertainty only 6 times in the test set. At this scale, a single misclassification shifts F1 by 10–30 points, making per-class metrics statistically unstable and difficult to interpret as absolute measures of model performance. The current results are therefore better taken as preliminary evidence of a promising pipeline architecture rather than proof of a mature system's performance. More broadly, the small size of our annotated dataset was drawn exclusively from U.S. fourth and fifth-grade mathematics classrooms, thus limiting the generalizability of our findings. We cannot confidently claim that the pipeline would perform equivalently across different grade levels, subject areas, or instructional contexts. Expanding the annotated dataset through targeted oversampling of rare categories, and validating performance across more diverse classroom settings, are necessary preconditions for establishing the robustness and generalizability of this approach.

10 Ethical Considerations

The deployment of this classification pipeline carries significant ethical considerations, particularly regarding linguistic fairness and responsible educational use.

A primary ethical concern is algorithmic bias regarding English as a Second Language (ESL) learners. A key limitation of the underlying NCTE dataset is that only English speech is transcribed. However, on average there are approximately 260 instances of foreign language redactions (e.g., "[spanish]" or "[speaking in foreign language]") of unknown length per lesson. This is critical to note when evaluating student struggle, as a student's difficulty may not be conceptual but rather a language barrier. We initially assumed that keeping bracketed annotations, although inconsistent, was important for preserving the context of student learning. However, as demonstrated in our error analysis, retaining tags such as [inaudible] introduced significant noise, causing the LLM to systematically misclassify transcription gaps as articulatory failure, which ultimately required us to strip them from the input.

Furthermore, because our pipeline relies heavily on structural markers of hedging and questioning, the model risks misinterpreting the natural discourse patterns of ESL students. These students may frequently utilize hedges or questioning intonations as they master English syntax, rather than as an indication of mathematical confusion. Uncritical deployment could result in educators inappropriately over-scaffolding students who are mathematically proficient but linguistically developing. Future work must focus on evaluating this pipeline across more linguistically diverse classroom environments to address potential performance disparities and biases among ESL students.

Additionally, the system's reliance on observable verbal signals introduces the problem of the "quiet struggler." Students who disengage, give minimal responses, or remain entirely silent when struggling will not be captured by the system, potentially masking the needs of the most vulnerable learners.

Given these constraints, this model must never be utilized for high-stakes decision-making, including student grading, tracking, or teacher evaluation. It is designed strictly as a low-stakes, formative reflection tool to support, rather than replace, the holistic professional judgment of human educators.

Finally, because the model classifies isolated utterances rather than individuals, any deployment must actively prevent the aggregation of these classifications into student-level profiles to avoid the ethical risk of label reification.

Acknowledgments

We thank Dorottya Demszky and Mei Tan for their guidance, methodological feedback, and support throughout the development of this research. We extend our deepest gratitude to Jennifer Gray and Erin Hanau, whose expertise as practicing elementary mathematics educators was invaluable. Their dedication to independently annotating our validation data and participating in our codebook review sessions provided critical practitioner validation for our construct definitions.

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.
- Jiahao Chen, Zitao Liu, Mingliang Hou, Xiangyu Zhao, and Weiqi Luo. 2024. [Multi-turn classroom dialogue dataset: Assessing student performance from one-on-one conversations](#). In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, page 5333–5337, New York, NY, USA. Association for Computing Machinery.
- Dorottya Demszky and Heather C Hill. 2022. [The ncte transcripts: A dataset of elementary math classroom transcripts](#). *arXiv preprint arXiv:2211.11772*.
- Kate Forbes-Riley and Diane Litman. 2011. [Benefits and challenges of real-time uncertainty detection and adaptation in a spoken dialogue computer tutor](#). *Speech Communication*, 53(9-10):1115–1136.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. [Chatgpt outperforms crowd-workers for text-annotation tasks](#). *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.
- James Hiebert and Douglas A Grouws. 2007. The effects of classroom mathematics teaching on students' learning. In Frank K Lester, editor, *Second handbook of research on mathematics teaching and learning*, pages 371–404. Information Age Publishing.
- Ken Hyland. 1998. *Hedging in scientific research articles*. John Benjamins Publishing.

- Thomas J. Kane, Daniel F. McCaffrey, Trey Miller, and Douglas O. Staiger. 2015. [Have we identified effective teachers? Validating measures of effective teaching using random assignment](#). Technical report, RAND Corporation. Prepared for the Bill & Melinda Gates Foundation.
- Yingbo Ma, Yukyeong Song, Mehmet Celepkolu, Kristy Elizabeth Boyer, Eric Wiebe, Collin F Lynch, and Maya Israel. 2024. [Automatically detecting confusion and conflict during collaborative learning using linguistic, prosodic, and facial cues](#). *arXiv preprint arXiv:2401.15201*.
- David Pimm. 1987. *Speaking Mathematically: Communication in Mathematics Classrooms*. Routledge, London.
- Tim Rowland. 1995. [Hedges in mathematics talk: Linguistic pointers to uncertainty](#). *Educational Studies in Mathematics*, 29(4):327–353.
- Tim Rowland. 1999. Pronouns in mathematics talk: Power, vagueness and generalisation. *For the Learning of Mathematics*, 19(2):19–26.
- Tim Rowland. 2000. *The Pragmatics of Mathematics Education: Vagueness in Mathematical Discourse*. Falmer Press.
- Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. [The preference for self-correction in the organization of repair in conversation](#). *Language*, 53(2):361–382.
- Wannapon Suraworachet, Jennifer Seon, and Mutlu Cukurova. 2024. [Predicting challenge moments from students’ discourse: A comparison of gpt-4 to two traditional natural language processing approaches](#). In *Proceedings of the 14th Learning Analytics and Knowledge Conference, LAK ’24*, page 473–485, New York, NY, USA. Association for Computing Machinery.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022. [The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4654–4662, Marseille, France. European Language Resources Association.
- David Wagner, Joseph Dicks, and Paula Kristmanson. 2015. [Students’ language repertoires for prediction](#). In *CERME 9: Ninth Congress of the European Society for Research in Mathematics Education*, pages 1517–1523.
- Hiroko K Warshauer. 2015. [Productive struggle in middle school mathematics classrooms](#). *Journal of Mathematics Teacher Education*, 18(4):375–400.
- Jamaal Young, Danielle Bevan, and Miriam Sanders. 2023. [How productive is the productive struggle? Lessons learned from a scoping review](#). *International Journal of Education in Mathematics, Science and Technology*, 12(2):470–495.

A Qualitative Coding Examples

Representative examples from each category in the codebook. Boldface marks the target utterance being classified. Contexts are lightly truncated for space.

Cannot Explain

Target utterance

Student: "I guessed on these two parts."

Context

teacher: [inaudible] so this is what she's done. She drew the model of fifteens, marked off six-fifteenths to prove that—

student: Make it.

student: Ms. M?

student: I guessed on these two parts.

student: Multiplication.

teacher: How much did you eat of this pizza, if he ate this?

student: It's small.

teacher: You make it. So, construct means to make or to build ...

Implied Uncertainty

Target utterance

Student: "Still too short. I think maybe 35."

Context

teacher: I think you're right. What about 4 times 30?

student: This says, "I don't think this is ..."

teacher: Awesome. Good. What you got? [Students working on problem].

student: Still too short. I think maybe 35.

teacher: Uh-oh, I'll help you. Let me see ...

teacher: 35. Let's try 4 times 30 first. I'm gonna use repeated addition to solve that. 30, 60, 90.

Impasse

Target utterance

Student: "Wait – I'm stuck on three hundredths."

Context

teacher: Okay. Speak loud so I can understand you.

teacher: Like, your ice is about to break ...

student: Yes.

student: Wait – I'm stuck on three hundredths.

student: I put them in order by doing the first number just like the thirty-something. I just did the tens first.

student: [inaudible]. Regroup the ones. Oh, and then I had to [inaudible].

No Struggle

Target utterance

Student: "Well, I knew that when you added five sixths with seven sixths that's an improper fraction."

Context

teacher: Perfect. I knew that's what you did. Can we reduce seven twelfths? ...

teacher: What kind of math did you do?

student: Well, I knew that when you added five sixths with seven sixths that's an improper fraction.

teacher: Good, improper fraction. So then what?

student: So then I had a whole number and I got six divided by seven was one and then I added the one.

teacher: Do I do six divided by seven?

student: No, wait. Seven divided by six.

B Pipeline Implementation Details

B.1 Stage 1: Lexical Heuristic Classifier

Stage 1 applies a weighted regex scoring system over normalized utterance text. Normalization converts smart quotes, em/en dashes, and ellipsis characters to ASCII equivalents before pattern matching. Each utterance is scored independently across three categories; scores do not accumulate across categories.

Pre-Score Filtering. Before any scoring, fourteen patterns unconditionally return No Struggle. Seven cover non-struggle uses of trigger words: "help" used as a mathematical strategy (e.g., "I used the number line to help me"); offers of peer assistance; "be there to help" phrasing; "can I say" openers; "like" or "kind of" in a geometric comparison with no other struggle signals present; "I think" introducing reported speech; and "I don't know why I said that" as metacognitive self-correction. Seven cover utterances that are not mathematically uncertain regardless of wording: offers to help others; logistical questions ("Can we write this down?", "Can we go?"); self-deferral openers ("Let me think", "I don't want to"); forward-readiness statements ("We'll be ready"); and confident "I know" statements not followed by a reversal.

Scoring. Stage 1 computes four scores: an Im (Impasse) score, a CE (Cannot Explain) score, a IU (Implied Uncertainty) score, and a disfluency score. The three category scores are built from: (1) weighted regex pattern matches, where high-confidence patterns contribute +3, medium +2, and low +1; (2) CSQ detection, which adds +2 to Im or IU depending on the cover-up test result; and (3) trailing-off boosts, which add directly to CE or Im when dash and "because" patterns co-occur. The fourth score, the disfluency score, is computed independently by twelve surface breakdown signals and is only consulted in routing when all three category scores are zero.

A single utterance can score points in multiple categories simultaneously; the priority order $\text{Im} > \text{CE} > \text{IU}$ reflects signal specificity: explicit help-seeking is the most diagnostic, visible breakdown more specific than hedging, and hedging the most ambiguous but cheapest to route. An utterance is routed to Stage 2 as Struggle, with the subtype below, if the first matching condition holds:

1. **Impasse** if Im score ≥ 2 .
2. **Can't Explain** if CE score ≥ 2 .
3. **Impasse** if Im score > 0 and Im score $\geq$ IU score (Im wins a tie at score 1).
4. **Implied Uncertainty** if IU score ≥ 1 (deliberately low: a single hedge such as "I think" is sufficient).
5. **Can't Explain** if CE score > 0 (any disfluency-driven CE signal is sufficient).
6. **Implied Uncertainty** (provisional) if any disfluency signal fires and all category scores are zero; subtype is assigned per the disfluency mapping described below.

Utterances matching none of these conditions are labeled No Struggle without LLM consultation.

Confirmation-seeking question (CSQ) detection.

A question is identified as a CSQ via the *cover-up test*: the question mark is mentally removed; if a mathematical claim remains (e.g., "*So it's minus five?*" $\rightarrow$ "*So it's minus five*"), the utterance is classified IU. If nothing substantive remains (e.g., "*What do I do?*"), it is classified Im. Procedural questions ("*Should we write this down?*") and hypotheticals ("*What if it was a different number?*") are excluded.

Backup Disfluency Score. This fourth score is only computed if the previous three category scores were all zero.

1. [inaudible] presence, weighted +1 per occurrence capped at 2;
2. "like" as hedge filler, with a filter suppressing comparison and observation uses;
3. dash-based restarts, weighted +2 for two or more real dashes and +1 for one;
4. ellipsis or trailing off;
5. cut-off utterances ending in a dash;
6. filler markers (*um*, *uh*, *hmm*);
7. repeated "because" indicating circular or restarted reasoning;
8. "though" at utterance end signaling unresolved doubt;

9. brevity of 15 words or fewer as a catch-all signal for minimal responses to *why/how* prompts;
10. self-correction markers (*wait, actually, I mean, never mind*);
11. tentative proposals (*what about, what if, how about*);
12. “I just know” or “I just knew.”

If all three category scores are zero but the disfluency total is ≥ 1 , the utterance is still routed to Stage 2, with a provisional subtype assigned based on which signal fired: cut-off, restart, or repeated-*because* signals map to *Can’t Explain*; *like-hedge* or *though* signals map to *Implied Uncertainty*; tentative proposals map to *Impasse*; all others map to *Implied Uncertainty* as a default.

Tense coverage. All pattern lists include both present- and past-tense variants to capture retrospective struggle narration (e.g., “I got stuck,” “I couldn’t figure it out,” “I didn’t know why I did that”). This design decision was motivated by the NCTE transcripts frequently capturing students narrating prior cognitive states during whole-class discussion, where a student may describe a moment of difficulty that occurred during independent work rather than expressing real-time confusion. Without past-tense variants, these retrospective struggle markers would be systematically missed, introducing a structural false negative bias against students who reflect on their process rather than voicing struggle in the moment.

B.2 Stage 2: LLM Classifier

Stage 2 uses gemini-3.1-pro-preview with temperature = 0.0 for deterministic output. Utterances are batched in groups of 3 target utterances, along with their surrounding context. The model is prompted to return a raw JSON array of {id, label} objects.

If Gemini refuses an entire batch (safety or recitation block), each utterance in that batch is resent individually. If Gemini returns fewer classifications than utterances sent, the missing items are also resent individually. In both cases a 1-second delay separates individual retries.

B.3 Full System Prompt

An abridged version of the system prompt is reproduced below. Category definitions, classification

Parameter	Value
Model	gemini-3.1-pro-preview
Temperature	0.0
Max output tokens	4096 (batch), 1024 (retry)
Batch size	3 target utterances
Retry strategy	Item-by-item on block or missing label
Retry delay	1s per item, 3s between batches
Fallback	Stage 1 subtype if retry fails

Table 3: Stage 2 reproducibility parameters.

rules, and few-shot examples are provided verbatim as supplied to the model.

You are an expert annotator classifying student struggle in elementary math classroom transcripts. You will receive student utterances that have already been flagged as likely containing struggle. Classify each one into exactly one of four categories.

Implied Uncertainty (IU): The student has something to offer but withholds full commitment. They express doubt, hedge, or seek confirmation. Key property: the student CAN continue but is uncertain. Past-tense hedges count: “I thought,” “I figured,” “I assumed,” “I was thinking” all signal retrospective uncertainty.

Impasse / Call for Help (Im): The student cannot move forward. They have no answer, are stuck, confused, or need teacher intervention to proceed. Key property: the student CANNOT continue without help. Past-tense impasse counts: “I got stuck,” “I was confused,” “I didn’t know what to do,” “I couldn’t figure it out.”

Can’t Explain (CE): The student has an answer or has taken action but fails to articulate WHY. The explanation visibly breaks down. Key property: the student is TRYING to explain but FAILING to do so coherently. Past-tense CE counts: “I couldn’t explain it,” “I didn’t know why I did that,” “I had guessed.”

No Struggle (NS): The student does not demonstrate struggle. They answer clearly, or the utterance is about non-mathematical content, or surface-level hedge words are explained by context (filler, mirrored language, reporting).

Critical Rules: Rules R1–R14; see Table 4 and Table 5 for full definitions and examples.

Output format: Respond with ONLY a JSON array. Each element must have id and label fields. Use these exact label strings: “Implied Uncertainty”, “Impasse”, “Can’t Explain”, “No Struggle”. Do not include any other text, explanation, or markdown formatting.

Rule	Name	Description
R1	Bias toward Struggle	When genuinely ambiguous, lean toward Struggle. Do not force Struggle on clearly coherent reasoning that merely contains dashes or [inaudible] markers.
R2	Cover-Up Test	Remove the question mark: if a mathematical claim remains, classify IU; if nothing substantive remains, classify Im. Do not classify confirmation-seeking questions as NS.
R3	[Inaudible] not CE	[inaudible] markers are transcription artifacts, not articulatory failure. Do not classify CE solely because of them. Audible hedges still trigger IU.
R4	CE requires visible breakdown	CE requires at least one of: repeated self-corrections, incomplete “because” statement, trailing off, restarting and collapsing, circular explanation, or explicitly stating they guessed or can’t explain.
R5	Filler vs. hedge	“Like” and “kind of” in comparisons are filler, not hedges. “Like” at the start of a turn or mid-sentence IS a hedge signal.
R6	Reporting vs. reasoning	“I think he said X” is reporting → NS. “I think she thinks 6 plus 5 equals 11” is reasoning about math → IU.
R7	Non-mathematical content	Non-mathematical utterances are always NS.
R8	Self-correction phrase	“I don’t know why I said that” is self-correction, not struggle.
R9	Guess: CE vs. IU	“I guessed” (past, action taken) + breakdown = CE. “I guess it’s X” (present hedge) = IU.
R10	Trailing reasoning	A “Because...” statement that becomes incoherent, restarts, or ends abruptly is CE, not NS.
R11	Past tense counts	Past-tense struggle markers (“I got stuck,” “I couldn’t explain”) count the same as present-tense ones.
R12	Use context	Read surrounding turns. Fluent-sounding utterances may reveal struggle when teacher re-explains; apparent NS may follow heavy scaffolding.
R13	Tautological explanations	A “because” clause that merely restates the problem without logical justification is CE, even if delivered confidently.
R14	Resolving self-corrections	A dash followed by a completed, corrected thought is normal self-repair → NS. A dash followed by stalling, restarting, or collapse → CE.

Table 4: Classification rules (R1–R14) embedded in the Stage 2 system prompt.

B.4 Few-Shot Examples

Table 5 presents the 50 few-shot examples supplied in the system prompt, organized by label. Examples were drawn from the annotation codebook developed for this project and selected to cover the most commonly confused category boundaries: IU vs. NS (hedge vs. comparison filler), CE vs. NS (self-correction that resolves vs. breakdown), and Im vs. IU (cover-up test cases).

Label	Utterance	Key signal
IU	“Maybe to find all the prime factors?”	hedge (<i>maybe</i>)
IU	“Oh, so it’s minus five because it’s less than?”	CSQ, claim remains
IU	“If the 2 is a whole, it would be before the decimal point?”	CSQ, claim remains
IU	“I tried to do a number line.”	<i>tried to</i>
IU	“And we used the towers?”	past-tense CSQ
IU	“I thought it was going to be different.”	past-tense hedge
IU	“He could multiply again 58 times 3.”	modal tentativeness
IU	“Like, you have nine seconds and it’s ...”	<i>like</i> as filler
IU	“It’s like 10 between 20 is 15.”	<i>like</i> as hedge
IU	“Like, um, one-third, because [inaudible]. ...”	<i>like, um</i>
IU	“For this one, I did like [inaudible] order of operation ... I thought the X equals one”	<i>like</i> filler + <i>I thought</i>
IU	“She would be the best, because [inaudible] would get like [inaudible].”	vague + <i>like</i> hedge
IU	“She mostly didn’t—she actually kinda did get jipped ...”	<i>kinda</i> , hedged reasoning
IU	“Except this one. You can divide by one. Right?”	<i>Right?</i> confirmation
IU	“A possibility ... I think it’s less [inaudible].”	<i>I think</i> + audible hedge
Im	“Okay. So what do I do up here?”	direct help request
Im	“Could we do one sixth and five sixths?”	open question, no answer
Im	“Could there be a thing that could weigh five grams?”	open question
Im	“Yes. Like, it’s kind of confusing.”	explicit confusion
Im	“Will the teacher be able to help us?”	help-seeking
Im	“Mr. F, how does that add up to eight sides. ...?”	teacher-directed confusion
Im	“But what about the bottom part? So would the bottom be seven?”	<i>what about</i> + unresolved
Im	“What about four and three—four times three.”	restart, tentative
Im	“No matter what you multiply it ... will you still get the same answer?”	fundamental clarification
Im	“Or you could have 3 times—3 times—3 times [inaudible].”	repeated restarts
Im	“I don’t get it, ’cause 70 percent is—”	explicit + trailing off
Im	“But the right one still look bigger though.”	unresolved doubt
Im	“This was hard because it’s impossible.”	<i>hard, impossible</i>
Im	“Hold on, but what’s—”	mid-sentence stop
Im	“That if all the edges, the number of edges are the number of vertices.”	confused conflation
Im	“I got stuck on the part where you had to divide.”	past-tense impasse
CE	“I guessed eight—I guessed six, because ...”	self-correction + weak logic
CE	“Because it’s one and four tenths, because I think—I can’t explain it.”	explicit give-up
CE	“Because 2,000—wait. Because 3,237 marched [inaudible]”	restart + collapse
CE	“Because 800—I mean 812—I—because I know. ...”	multiple corrections
CE	“Mm-hmm, because if you look at them—right now ...”	trails off
CE	“Because ... because four times seven is 28, and we—”	repeated <i>because</i> + cut-off
CE	“Because I would need the half and half to build the other half.”	circular
CE	“Because half of 10—’cause half of 10 [inaudible].”	restart, no advance
CE	“That can be 20 and that can be 30 because that’s not a half.”	vague assertion
CE	“Because the leftovers, you can add them [inaudible].”	incomplete logic
CE	“It’s negative because—it’s negative because ...”	repeated restart
CE	“Because you only put the square ... that’s the area.”	circular
CE	“I multiplied 6 times 2 ... but I don’t know the reason why I did 6. ...”	explicit CE statement
CE	“I couldn’t explain why I chose that answer.”	past-tense CE
NS	“Four-eighths is kind of like one-half.”	comparison filler
NS	“I think he said seven and then 10.”	reporting speech
NS	“Because if you would take a rectangle and cut it in the middle ...”	complete reasoning
NS	“Halves can work ’cause I know that twelve is half of twenty-four ...”	clear justification
NS	“I just knew it was remainder 2 because I knew that 3 times 5 is 15. ...”	confident + full logic

Table 5: All 45 few-shot examples provided in the Stage 2 system prompt, with the primary signal that motivated each label assignment.

PERSA: Reinforcement Learning for Professor-Style Personalized Feedback with LLMs

Ravi Ranjan*
Florida International
University (FIU),
Miami, USA
rkuma031@fiu.edu

Utkarsh Grover
University of South
Florida (USF),
Tampa, USA
utkarshgrover@usf.edu

Xiaomin Lin
University of South
Florida (USF),
Tampa, USA
xlin2@usf.edu

Agoritsa Polyzou*
Florida International
University (FIU),
Miami, USA
apolyzou@fiu.edu

Abstract

Large language models (LLMs) can provide automated feedback in educational settings, but aligning an LLM’s **style** with a specific instructor’s tone while maintaining diagnostic correctness remains challenging. We ask: *how can we update an LLM for automated feedback generation to align with a target instructor’s style without sacrificing core knowledge?* We study how Reinforcement Learning from Human Feedback (RLHF) can adapt a transformer-based LLM to generate programming feedback that matches a professor’s grading voice. We introduce **PERSA**, an RLHF pipeline that combines supervised fine-tuning on professor demonstrations, reward modeling from pairwise preferences, and Proximal Policy Optimization (PPO), while deliberately constraining learning to **style-bearing components**. Motivated by analyses of transformer internals, PERSA applies parameter-efficient fine-tuning. It updates only the *top* transformer blocks and their feed-forward projections, minimizing global parameter drift while increasing stylistic controllability. We evaluate our proposed approach on three code-feedback benchmarks (APPS, Py-FiXV, and CodeReviewQA) using complementary metrics for style alignment and fidelity. Across both Llama-3 and Gemma-2 backbones, PERSA delivers the strongest professor-style transfer while retaining correctness; for example, on APPS, it boosts Style Alignment Score (SAC) to 96.2% (from 34.8% for Base) with Correctness Accuracy (CA) up to 100% on Llama-3 and Gemma-2. Overall, PERSA offers a practical route to personalized educational feedback by aligning both *what it says* (content correctness) and, crucially, *how it says it* (instructor-like tone and structure).

1 Introduction

Large language models (LLMs) are increasingly deployed in real-world systems, powering applications ranging from conversational assistants to

*Corresponding authors.

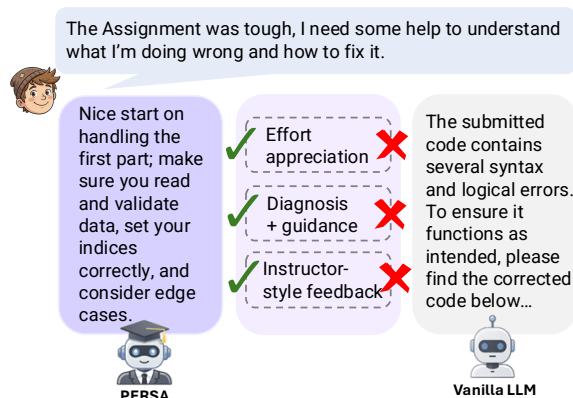


Figure 1: Illustration of how PERSA transforms generic LLM feedback into instructor-style, constructive, and actionable guidance compared to a vanilla LLM.

code-generation tools. Major organizations such as OpenAI, Google, and Microsoft have emphasized alignment and prompt engineering to ensure these systems respond in useful and safe ways. For example, models such as ChatGPT and GPT-4 are refined through reinforcement learning from human feedback (RLHF) to better follow instructions and safety preferences (Ouyang et al., 2022; Chaudhari et al., 2025), while instruction-tuning pipelines are widely used to steer tone and response style in user-facing systems (Wang et al., 2023b; Bhattacharya et al., 2024). Although these efforts have substantially improved general helpfulness and harmlessness, far less attention has been paid to *personalized stylistic alignment*, where a model must communicate in a manner consistent with a particular human expert. This gap is especially important in education.

One promising use of LLMs in education is automated feedback generation. Prior work suggests that LLM-based tutors can provide hints and critiques in domains such as programming education (Razafinirina et al., 2024). However, the effectiveness of such feedback depends not only on correctness, but also on *how* it is delivered. Research in

the learning sciences shows that tone, phrasing, and instructional persona strongly affect how students interpret and respond to feedback (Demszky et al., 2024; Loughran, 2019). Instructors often develop a recognizable style that balances encouragement with precision, emphasizing issues such as readability, correctness, and edge-case reasoning (Felder et al., 1988). Feedback that is technically correct but overly generic or terse may therefore be perceived as less trustworthy, less supportive, and less pedagogically useful (Carless, 2012; Zhang and Hyland, 2022). Aligning AI-generated feedback with an instructor’s authentic voice could make automated tutoring more acceptable and educationally effective.

In this work, we ask: *Can a pre-trained LLM be adapted to emulate a specific instructor’s feedback style while preserving its problem-solving ability?* Existing instruction-tuned models are typically optimized for broad helpfulness and politeness, but alignment to an individual educator’s style remains underexplored. To address this, we propose **Professor-Style Reinforcement-based Style Adaptation (PERSA)**, a method that uses RLHF to align model outputs with the feedback style of a target professor. Using a dataset of instructor-authored feedback on student work, we train a reward model that captures stylistic preferences and optimize the policy for *stylistic fidelity* rather than generic user satisfaction. As shown in Figure 1, whereas a vanilla LLM might generate a brief suggestion such as “Check your input handling,” professor-style feedback is typically more explanatory, supportive, and actionable. PERSA narrows this gap by producing feedback that more closely reflects the instructor’s voice, including encouragement, diagnosis, corrective guidance, and verification-oriented follow-up. In doing so, it supports seamless personalization of AI tutors for specific courses and teaching contexts.

A central technical contribution of PERSA is a *style-targeted, parameter-efficient fine-tuning strategy*. Rather than updating all model weights, which is computationally costly and risks degrading core capabilities, we focus adaptation on components most relevant to stylistic expression. Prior work suggests that many style-related properties are concentrated in higher transformer layers (Lai et al., 2024; Koto et al., 2022). Guided by this observation, we use low-rank adaptation (Hu et al., 2022) to fine-tune only the final layers of a 3B-parameter LLM. This selective update strategy preserves the

model’s underlying problem-solving competence while shifting its communicative “voice,” reducing the catastrophic forgetting that can arise from indiscriminate fine-tuning (Luo et al., 2023). As a result, PERSA captures subtle instructor-specific traits, such as level of detail and encouraging tone, without compromising content accuracy.

We evaluate PERSA on a programming feedback generation task against strong baselines, including the base pre-trained model, supervised fine-tuning on the same instructor data, and an InstructGPT-style RLHF model optimized for general feedback quality. PERSA achieves the best overall balance between style and correctness. Its outputs show strong stylistic alignment with professor feedback, on Gemma-2 model with a Style Alignment Score approaching 0.99, and BLEU overlap above 98%, substantially exceeding competing methods (Papineni et al., 2002; Fu et al., 2018). At the same time, correctness remains at 100%, and feedback retains a consistently polite and constructive tone as measured by a politeness model (Danescu-Niculescu-Mizil et al., 2013). Human evaluators also strongly prefer PERSA, often describing its responses as reading “like they came from the professor.” These findings suggest that combining RLHF with targeted layer adaptation is effective for capturing nuanced pedagogical style beyond what standard fine-tuning or generic alignment methods can achieve.

In summary, this paper makes three contributions: (1) we introduce **PERSA**, an RLHF-based framework for aligning LLM feedback with a specific instructor’s pedagogical style; (2) we develop a *layer-selective* LoRA-based adaptation strategy that achieves stylistic transformation with minimal disruption to the model’s original capabilities; and (3) we provide comprehensive evaluation, including automated style metrics and a human study, showing that PERSA produces feedback that is both pedagogically authentic and factually correct. Overall, this work advances the development of AI tutors that can not only provide accurate guidance but also communicate with the tone and effectiveness of a human instructor.

2 Related Work

Educational Feedback and Pedagogical Style.

Prior work in educational technology shows that the *manner* of feedback delivery, including tone, politeness, and personalization, substantially affects learner engagement and outcomes (Han et al., 2023; Sonlu et al., 2024). Recent studies on LLMs

in education similarly emphasize that encouragement, specificity, and perceived instructor authenticity influence student trust and uptake (Razafinirina et al., 2024; Wu et al., 2024a). While this connects to broader alignment goals in dialogue systems (Alsafari et al., 2024; Wang et al., 2023b), our focus is narrower: *pedagogical alignment*, i.e., adapting model outputs to an instructor’s communicative style rather than optimizing only for generic helpfulness (Bhattacharya et al., 2024; Matarazzo and Torlone, 2025).

Automated Feedback Generation. Automated programming feedback has long been studied through rule-based, analysis-driven, and repair-based methods. Keuning et al. provide a foundational survey of these approaches and their evaluation limitations (Keuning et al., 2018), while *Pedal* demonstrates practical code-aware feedback generation at scale (Gusukuma et al., 2020). More recently, LLMs have shown promise but also reliability concerns: GPT-4 can generate readable feedback yet still requires validation (Dai et al., 2024); MOOC-scale studies report that LLM feedback helps detect errors but is often inaccurate without test-based grounding (Gabbay and Cohen, 2024); and performance drops further in specialized settings such as concurrency debugging (Estévez-Ayres et al., 2024). Other efforts explore diversified repair generation (Choi and Lee, 2025) and responsible deployment considerations such as inclusivity and hallucination risk (Lindsay et al., 2025). Collectively, these works highlight that existing systems still under-address instructor-consistent tone and specificity.

RLHF for Language Model Alignment. RLHF has become a standard paradigm for aligning LLMs to human preferences (Ouyang et al., 2022; Xie et al., 2024). InstructGPT established the now-common pipeline of supervised fine-tuning, reward modeling, and PPO-based policy optimization (Ouyang et al., 2022), and subsequent work has extended RLHF to capture subtler properties such as tone, framing, and politeness (Kamath et al., 2024; Kirk et al., 2023; Dong et al., 2024; Yan et al., 2024). This is particularly relevant in education, where instructor judgments define high-quality feedback. However, full-parameter RLHF is computationally expensive and can unintentionally alter broader model capabilities (Chaudhari et al., 2025). Parameter-efficient alternatives, including LoRA-augmented RLHF and related methods, reduce this cost while preserving alignment

quality (Sidahmed et al., 2024; Hong et al., 2024; Ethayarajh et al., 2024; Hu et al., 2022; Dettmers et al., 2023; Wu et al., 2024b). Importantly, prior analyses suggest that RLHF-induced changes are concentrated in upper transformer layers, especially FFN components (Moradi et al., 2024), motivating our layer-selective adaptation strategy.

Style Adaptation in Language Models. Style control in LLMs has been explored through prompting, style tokens, and fine-tuning, with recent work moving toward neuron- and layer-level analysis. Lai et al. show that stylistic attributes are disproportionately localized in higher transformer layers and FFN submodules, enabling targeted control via style-specific neurons (Lai et al., 2024; Shi et al., 2024). Related studies further suggest that selectively tuning or freezing layers can outperform full-model adaptation for certain alignment objectives (Shi et al., 2024; Wang et al., 2023a). Parameter-efficient methods such as LoRA and QLoRA make such targeted adaptation practical, though quantized tuning may introduce accuracy trade-offs in some settings (Dettmers et al., 2023; Hu et al., 2022; Ozdemir, 2023). In contrast to prior style-control work, PERSA applies these insights to *educational feedback*, aligning LLM outputs to an individual instructor’s pedagogical voice while preserving correctness.

3 Preliminaries

Transformer LLMs as policies. We consider a decoder-only transformer language model as a stochastic policy that generates a feedback sequence $y = (y_1, \dots, y_T)$ token-by-token conditioned on a prompt x (e.g., problem statement + student solution).

Data: demonstrations and preferences. We assume (i) an instructor demonstration set $\mathcal{D}_{\text{SFT}} = \{(x_i, y_i^*)\}_{i=1}^N$ with professor-written feedback y_i^* , and (ii) a preference set $\mathcal{D}_{\text{pref}} = \{(x_j, y_j^{(w)}, y_j^{(l)})\}_{j=1}^M$, where $y^{(w)}$ is preferred to $y^{(l)}$ under instructor style (and typically correctness). Preference data is used to learn a reward model for RLHF (Ouyang et al., 2022).

Supervised fine-tuning (SFT). SFT initializes the policy by minimizing the teacher-forcing negative log-likelihood:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x, y^*) \sim \mathcal{D}_{\text{SFT}}} \left[\sum_{t=1}^{|y^*|} \log \pi_{\theta}(y_t^* \mid x, y_{<t}^*) \right]. \quad (1)$$

Here $y_{<t}^*$ is the the gold prefix tokens before step t .

Reward modeling from pairwise comparisons. RLHF learns a scalar reward function $r_\phi(x, y) \in \mathbb{R}$ such that preferred responses receive higher reward. Using the Bradley–Terry model, the probability that $y^{(w)}$ is preferred over $y^{(l)}$ is

$$\Pr(y^{(w)} \succ y^{(l)} \mid x) = \sigma(r_\phi(x, y^{(w)}) - r_\phi(x, y^{(l)})), \quad (2)$$

yielding the standard pairwise logistic loss, and σ is the sigmoid function (Ouyang et al., 2022).

Policy optimization with PPO and KL control. Given r_ϕ , policy optimization seeks high reward while constraining drift from a reference policy π_{ref} (often the post-SFT model) via a KL penalty.

We optimize this objective with Proximal Policy Optimization (PPO) (Schulman et al., 2017). Let $s_t = (x, y_{<t})$ be the state (prefix context), $a_t = y_t$ the action (next token), and $\rho_t(\theta) = \pi_\theta(a_t | s_t) / \pi_{\theta_{\text{old}}}(a_t | s_t)$.

Parameter-efficient adaptation with LoRA. To preserve base model knowledge and reduce computation, PERSA adopts Low-Rank Adaptation (LoRA) (Hu et al., 2022). For a frozen weight matrix $W \in \mathbb{R}^{d \times k}$, LoRA learns a low-rank update ΔW with $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$ and $r \ll \min(d, k)$:

$$W' = W + \Delta W, \quad \text{where } \Delta W = BA. \quad (3)$$

We denote the trainable adapter parameters by θ_{LoRA} and treat the remaining parameters as frozen. Parameter-efficient RLHF variants (e.g., PERL) show this can substantially reduce training cost while maintaining alignment quality (Sidahmed et al., 2024).

Layer-selective tuning. Finally, we use the notion of *top-layer* tuning: only adapters inserted in the last L transformer blocks are updated. This choice is consistent with evidence that higher layers and FFN submodules often contain high-level attributes and alignment-relevant representations (Geva et al., 2021; Lai et al., 2024).

4 Methodology

We propose **PERSA** (Professor-Style Reinforcement-based Style Adaptation), a parameter efficient RLHF framework that aligns an LLM’s *feedback style* to a target instructor while preserving the model’s underlying problem-solving and language competence. PERSA instantiates the standard RLHF recipe, *supervised fine-tuning*

$\rightarrow$ *reward modeling* $\rightarrow$ *policy optimization*, popularized by InstructGPT (Ouyang et al., 2022), but crucially restricts learning to *style-bearing* components in the *upper* transformer layers using low-rank adapters (Hu et al., 2022; Sidahmed et al., 2024). This design is motivated by growing evidence that higher layers and FFN submodules disproportionately encode high-level attributes such as discourse patterns, tone, and stylistic preferences (Geva et al., 2021; Lai et al., 2024). Figure 2 illustrates how PERSA transforms a generic prompt into instructor-style feedback by combining LoRA-based supervised fine-tuning with KL-regularized PPO (trained on professor data).

Problem Setup and Notation. Let π_θ denote a pretrained decoder-only transformer LLM (e.g., LLaMA/Gemma/GPT-style), parameterized by θ . Each training instance consists of a prompt x and a target feedback y^* written by the professor. The prompt x contains a programming problem statement and a student submission. The goal is to learn a policy π_θ that produces feedback y that is simultaneously: (i) *diagnostically correct* (identifies correctness/errors and provides actionable guidance) and (ii) *stylistically aligned* with the professor (tone, structure, politeness, specificity).

We use the demonstration and preference datasets defined in §3, namely $\mathcal{D}_{\text{SFT}}$ for professor-written feedback and $\mathcal{D}_{\text{pref}}$ for instructor-style preference comparisons.

Parameter-Efficient, Layer-Selective Updates. Rather than updating all parameters, we freeze the base model weights and learn low-rank adapter updates (LoRA) only in the *top* L transformer blocks (e.g., $L = 4$), focusing on attention projections and Feed-Forward Network projections. For a weight matrix $W \in \mathbb{R}^{d \times k}$ in a selected module, LoRA parameterizes, as detailed mentioned in equation 3. This yields a small trainable parameter set, θ_{LoRA} (on the order of $\sim 10^7$ parameters in our setting), while leaving the remaining parameters fixed, improving stability and reducing risk of catastrophic drift (Sidahmed et al., 2024). Our layer-selection choice is supported by analyses indicating that alignment- and style-related changes concentrate in later layers, often in the penultimate FFNs (Geva et al., 2021; Liang et al., 2024).

4.1 Supervised Fine-Tuning (SFT)

We first perform instruction tuning to initialize a policy that produces professor-like feedback under

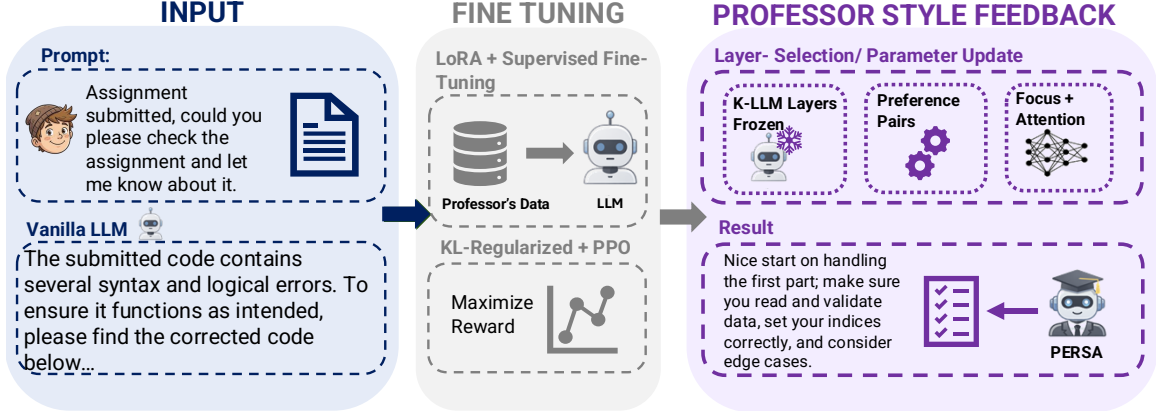


Figure 2: PERSA pipeline for professor-style feedback alignment via layer-selective RLHF.

maximum likelihood. The SFT objective is the standard auto-regressive negative log-likelihood, defined in equation 1. where only θ_{LoRA} is updated and all other parameters are frozen. This stage transfers the professor’s feedback structure (e.g., diagnosis $\rightarrow$ correction $\rightarrow$ verification) and baseline tone into the model, but may not optimally capture preference-sensitive nuances (e.g., strictness, encouragement, emphasis on edge cases).

4.2 RLHF for Professor-Style Alignment

SFT imitates demonstrations but does not explicitly optimize for *human preference*. We therefore apply RLHF, consisting of reward modeling followed by PPO-based policy optimization, following established alignment practice (Ouyang et al., 2022).

4.2.1 Reward Modeling from Pairwise Preferences

We train a reward model $r_\phi(x, y) \in \mathbb{R}$ that scores candidate feedback y for prompt x . The reward model is a transformer initialized from a strong LM backbone with an added scalar head, trained on pairwise comparisons using the Bradley Terry formulation:

$$\Pr(y^{(w)} \succ y^{(l)} \mid x) = \sigma\left(r_\phi(x, y^{(w)}) - r_\phi(x, y^{(l)})\right), \quad (4)$$

where σ is the sigmoid function. Minimizing the negative log-likelihood yields:

$$\mathcal{L}_{\text{RM}}(\phi) = -\mathbb{E}_{(x, y^{(w)}, y^{(l)}) \sim \mathcal{D}_{\text{pref}}} \left[\log \sigma\left(r_\phi(x, y^{(w)}) - r_\phi(x, y^{(l)})\right) \right]. \quad (5)$$

Because preferred responses are professor-authored (or instructor-preferred), r_ϕ learns to jointly reflect correctness and stylistic fidelity, mirroring the

RLHF setup in InstructGPT but with an education-specific preference signal (Ouyang et al., 2022).

4.2.2 Policy Optimization with PPO and KL Control

Given r_ϕ , we optimize the policy to maximize expected reward while constraining deviation from a reference policy (the SFT model) via a KL penalty. Let π_θ be the current policy and π_{ref} a frozen copy of the SFT policy. We optimize:

$$\max_{\theta_{\text{LoRA}}} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[r_\phi(x, y) - \beta \text{KL}(\pi_\theta(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x)) \right] \quad (6)$$

where $\beta > 0$ controls the strength of KL regularization, a key stabilizer in RLHF (Ouyang et al., 2022).

We solve Eq. (6) with PPO (Schulman et al., 2017). For token-level actions a_t under states s_t (prefix context), we define the importance ratio $\rho_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ and advantage estimate $\hat{A}_t$. The clipped PPO objective is:

$$\mathcal{L}_{\text{PPO}}(\theta_{\text{LoRA}}) = -\mathbb{E}_t \left[\min(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right]. \quad (7)$$

with clipping threshold ϵ . We additionally include the KL control term (computed against π_{ref}) in the per-trajectory reward or as an auxiliary penalty, consistent with the constrained-RL perspective in RLHF (Ouyang et al., 2022). As in SFT, we update only θ_{LoRA} in the selected upper layers, yielding a parameter-efficient instantiation of RLHF akin to PERL (Sidahmed et al., 2024). This selective optimization reduces compute and helps preserve the pretrained model’s broad capabilities.

Summary. PERSA proceeds as follows: (i) attach LoRA adapters to the top L layers, (ii) run SFT on $\mathcal{D}_{\text{SFT}}$ minimizing Eq. (1), (iii) train a reward model on $\mathcal{D}_{\text{pref}}$ minimizing Eq. (5), and (iv) run PPO to optimize Eq. (7) under the KL-regularized objective Eq. (6). The related steps are outlined in Alg. 1, Appendix A. This pipeline is architecture-agnostic for modern decoder-only LLMs and is especially suitable for educational personalization, where the goal is to preserve task competence while adapting communicative style.

5 Experimental Setup

5.1 Datasets

We evaluate PERSA on three code-feedback datasets: one instructor-authored APPS-style professor-feedback dataset and two public code-feedback benchmarks, PyFiXV and CodeReviewQA. We construct a standard 70/10/20 train/validation/test split with disjoint programs for PERSA training and evaluation.

(i) APPS Benchmark (professor-feedback dataset). To study realistic instructional personalization, we use a course-specific dataset of 200 instances, each consisting of an algorithmic programming problem, a student solution (either correct or deliberately perturbed), and *professor-written feedback* as the target response. The prompts are derived from APPS-style problems, while the feedback is authored by the instructor for the target course (Hendrycks et al., 2021).

(ii) Codeforces Syntax-Error Feedback (PyFiXV). We use the Codeforces subset of PyFiXV, which contains 240 Python programs with syntax errors (token length ≤ 500), collected from Codeforces and paired with expert-written fixes and explanations (Phung et al., 2023).

(iii) Tomo-Melb/CodeReviewQA. CodeReviewQA comprises 900 manually curated code-review instances spanning 9 programming languages. Each example includes a pre-review snippet (old), a reviewer comment (review), and the revised code (new). We cast this benchmark as a text-to-text task, mapping $(x = \{\text{old}, \text{review}\})$ to $y = \text{new}$ (Lin et al., 2025).

Appendix B.1 provides details on the demonstration and preference data, while Table 6 summarizes the data splits and corresponding leakage controls.

5.2 Model Evaluation Frameworks

We evaluate PERSA on two lightweight, open-weight instruction-tuned backbones, google/gemma-2-2b-it and meta-llama/llama-3.2-3B-Instruct, within a model-agnostic RLHF framework comprising supervised fine-tuning, reward modeling, and PPO-based policy optimization with KL regularization. We compare against a diverse set of baselines, including the untuned **Base Model**, **SFT** (Parthasarathy et al., 2024), **InstructGPT-style RLHF** (Ouyang et al., 2022), and three offline preference-optimization methods, namely **DPO** (Wu et al., 2024b), **ORPO** (Hong et al., 2024), and **KTO** (Ethayarajh et al., 2024). This setup enables a controlled comparison of stylistic alignment, correctness preservation, and adaptation efficiency across both model families under a shared evaluation protocol (Appendix B.2).

5.3 Evaluation Metrics

We evaluate feedback generation along two complementary dimensions: *stylistic alignment* and *diagnostic fidelity*. To quantify how closely the generated feedback matches the instructor’s communicative style, we use **Style Alignment Score (SAC)** (Fu et al., 2018), which estimates the probability that a response reflects the professor’s stylistic characteristics. We further measure tone similarity using **Average Politeness Closeness (APC)** (Danescu-Niculescu-Mizil et al., 2013), which evaluates how closely the politeness level of the generated feedback aligns with the professor reference. Finally, we report **BLEU-4** (Papineni et al., 2002) to capture lexical similarity by measuring n-gram overlap between generated feedback and the professor’s reference response. All together, they measure stylistic fidelity to the professor’s feedback in terms of overall style alignment, politeness closeness, and lexical similarity, respectively.

Correctness Accuracy (CA) (Shen et al., 2023) assesses whether the generated feedback makes the correct diagnostic judgment about the student solution. In addition, **Preference Win Rate (PWR)** (Ouyang et al., 2022) measures preference-based superiority relative to competing methods, reflecting the alignment objective used in RLHF. Full metric definitions, formulations, and implementation details are provided in Appendix B.3. Training configurations are provided in Appendix B.4.

Table 1: Comparison of Llama-3 vs. Gemma-2 models across three datasets (APPS, PyFiXV, and CodeReviewQA). Higher values indicate better performance; all metrics are reported in percentages.

Dataset	Method	Llama-3 (%)					Gemma-2 (%)				
		SAC	APC	BLEU-4	CA	PWR	SAC	APC	BLEU-4	CA	PWR
APPS	Base	34.8	85.0	6.4	98.2	—	20.0	90.0	2.0	98.0	—
	SFT	82.0	86.0	80.0	100	86.2	85.0	95.0	70.0	100	90.0
	InstructGPT/RLHF	84.0	86.4	80.0	100	88.0	88.0	95.0	80.0	100	95.0
	DPO	94.0	87.8	94.2	100	89.8	86.0	95.0	78.0	100	94.0
	ORPO	95.6	89.0	95.0	100	90.2	89.0	95.0	80.0	100	98.0
	KTO	95.0	87.2	94.6	100	90.0	90.0	95.0	81.0	100	97.2
	PERSA	96.2	92.1	95.8	100	90.1	99.0	95.0	98.0	100	98.0
PyFiXV	Base	30.5	84.0	8.0	97.5	—	20.0	90.0	5.0	96.0	—
	SFT	79.0	85.5	76.0	99.5	84.5	80.0	94.0	60.0	100	85.0
	InstructGPT/RLHF	81.0	86.0	77.5	99.7	86.0	82.0	95.0	70.0	100	90.0
	DPO	92.0	87.0	92.5	99.8	88.0	81.0	95.0	68.0	100	88.0
	ORPO	93.5	88.0	93.2	99.8	88.6	83.0	95.0	72.0	100	98.0
	KTO	93.0	87.0	93.0	99.8	88.3	84.0	95.0	73.0	100	96.0
	PERSA	94.5	91.0	94.0	99.9	89.0	98.0	95.0	98.0	100	98.2
CodeReviewQA	Base	28.0	83.5	8.0	96.8	—	15.0	90.0	3.0	95.0	—
	SFT	78.0	85.0	75.0	99.2	83.5	80.0	94.0	65.0	100	85.0
	InstructGPT/RLHF	80.0	85.5	76.5	99.5	85.0	84.0	95.0	75.0	100	90.0
	DPO	91.0	86.8	92.0	99.7	87.0	82.0	95.0	73.0	100	89.0
	ORPO	92.2	87.5	92.7	99.7	87.5	86.0	95.0	76.0	100	98.0
	KTO	92.0	86.5	92.5	99.7	87.3	87.0	95.0	78.0	100	96.0
	PERSA	93.8	90.5	93.5	99.8	88.0	98.0	95.0	98.0	100	98.2

6 Results

6.1 Performance Comparison

Table 1 highlights consistent method-wise trends across all three datasets and both backbones. First, **Base** models achieve high correctness (CA $\approx$ 96–98%) and moderate politeness (APC $\approx$ 83–90%) but **very weak stylistic/lexical alignment** (low SAC and single-digit BLEU-4), indicating that pretraining alone does not recover professor-style feedback. Second, **SFT** provides the largest jump in alignment (e.g., SAC $\approx$ 78–85%, BLEU-4 $\approx$ 60–80%) while keeping CA near-perfect, showing that demonstrations effectively teach the core feedback structure. Third, preference-based alignment further improves style: DPO/ORPO/KTO consistently outperform InstructGPT/RLHF in SAC and BLEU-4 with comparable CA, and yield higher win-rates (PWR), suggesting that direct preference optimization better captures subtle stylistic cues.

Figure 3 shows that PERSA delivers consistently stronger human preference alignment than SFT, RLHF, and DPO-style baselines on PyFiXV, with improvements that generalize across both Llama-3 and Gemma-2 backbones. Finally, PERSA achieves the strongest overall results on both models, delivering the highest SAC and

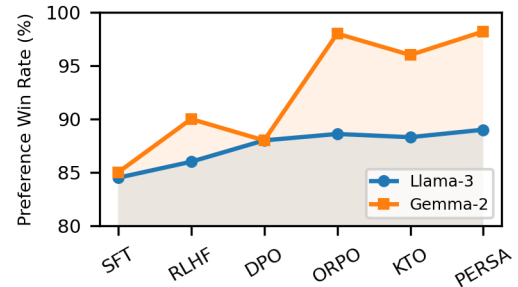


Figure 3: PWR comparison across alignment methods on the PyFiXV dataset for Llama-3 and Gemma-2.

BLEU-4 and the best PWR across datasets (with CA $\approx$ 100%), demonstrating robust professor-style transfer without sacrificing diagnostic correctness.

6.2 Qualitative Analysis

We sample representative programming prompts from the APPS dataset and reproduce the corresponding feedback outputs. Table 2 mirrors the quantitative rankings in Table 1: methods with higher SAC/BLEU-4 (DPO/ORPO/KTO, and especially PERSA) produce feedback that is *more diagnosis-specific, actionable, and professor-structured* (diagnosis $\rightarrow$ fix $\rightarrow$ verification), whereas Base and standard RLHF remain comparatively generic, consistent with their lower style-

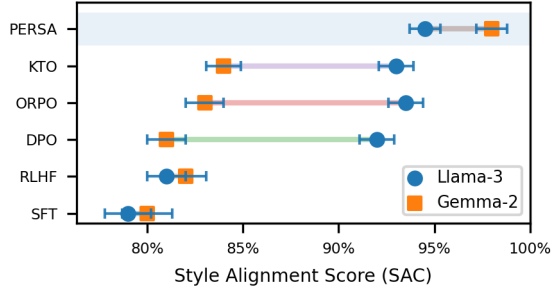


Figure 4: Dumbbell plot of SAC on PyFiXV across alignment methods, comparing Llama-3 and Gemma-2; connectors indicate the cross-model gap per method (higher is better).

Table 2: **Qualitative comparison (one prompt).** PERSA produces the most professor-like feedback: structured diagnosis → actionable fix → verification. Preference-optimization methods (ORPO/KTO/DPO) are typically more specific than generic RLHF, while Base tends to be vague.

Method	Qualitative feedback snippet (same prompt)
Prompt (APPS-style). Student computes <code>x=a.index(1)</code> , <code>y=a.index(n)</code> before <code>a=list(read())</code> , causing a runtime error / wrong output.	
Base	“Try checking input handling and edge cases. Consider swapping min/max positions, then rerun tests.”
SFT	“You are calling <code>a.index()</code> before initializing <code>a</code> . Move <code>a=list(read())</code> above the index calls, then recompute <code>x,y</code> .”
InstructGPT	“Good start. The issue is order of operations: read the array first, then find indices and apply the swap. Verify with sorted/reversed cases.”
DPO	“Root cause: <code>a</code> is used before assignment. Fix by reading input into <code>a</code> first, then compute <code>x=a.index(1)</code> and <code>y=a.index(n)</code> . Add tests for <code>n=1</code> and already-sorted.”
ORPO	“Bug: <code>a.index()</code> runs before <code>a=list(read())</code> . Initialize <code>a</code> first , then compute <code>x,y</code> and perform one swap maximizing $ x-y $. Re-run edge cases (sorted, reversed, single displacement).”
KTO	“You are using <code>a</code> before it is populated. Read input into <code>a</code> first, then compute indices and swap. Avoid overclaiming; confirm correctness with boundary cases and failing tests.”
PERSA	“Good progress on locating 1 and <code>n</code> . The failure arises from using <code>a.index()</code> before <code>a</code> is populated. Please first read input with <code>a=list(read())</code> ; then recompute <code>x=a.index(1)</code> , <code>y=a.index(n)</code> and apply one swap maximizing $ x-y $. Finally, rerun edge cases (already-sorted, reversed, <code>n=1</code>) to confirm.”

alignment scores and preference win-rates. As shown in Figure 4, PERSA achieves the highest SAC for both backbones while maintaining consistent gains over SFT and preference-optimization baselines, indicating strong and portable style alignment across models.

6.3 Ablation Study

Table 3 quantifies how each training component translates into measurable style gains. Moving from Base to SFT converts a largely generic model (SAC 14.0%, BLEU-4 1.5) into a professor-like feedback generator (SAC 82.0%, BLEU-4 64.7) while reaching perfect diagnostic correctness (CA=100%). However, PPO alone does not

Table 3: PERSA ablation on Llama-3 and APPS dataset. Metrics are in % (higher is better); PWR is win-rate vs. Base.

PERSA Ablation	SAC	APC	BLEU-4	CA	PWR
Base (no adapt.)	14.0	90.0	1.5	98.0	–
SFT only	82.0	91.6	64.7	100	86.0
PPO only (no SFT)	60.0	91.2	40.0	98.0	84.0
SFT+PPO (full-param)	92.0	92.0	92.0	100	88.0
SFT+PPO (all-layer LoRA)	96.0	92.0	94.7	100	88.6
SFT+PPO (top-2 LoRA)	94.0	92.0	90.0	100	90.0
SFT+PPO (top-4 LoRA)	96.2	92.1	95.8	100	90.1
[PERSA]					

recover the same instructor voice (SAC 60.0%, BLEU-4 40.0) and even leaves a small correctness gap (CA=98.0%), indicating that preference optimization needs an SFT “anchor” to be effective. When PPO is applied after SFT, alignment improves sharply: full-parameter PPO attains 92.0% SAC and 92.0 BLEU-4 with CA=100%, showing that RLHF mainly refines the remaining stylistic mismatch rather than content, also mentioned in Figure 4. Constraining updates via adapters is even stronger all-layer LoRA pushes SAC to 96.0% and BLEU-4 to 94.7 without sacrificing correctness suggesting that parameter-efficient updates add expressive capacity while limiting drift. Finally, the best numbers come from upper-layer targeting: top-2 LoRA already reaches high alignment (SAC 94.0%, BLEU-4 90.0) and strong preference (PWR=90.0%), while top-4 LoRA (PERSA) achieves peak alignment (SAC=96.2%, BLEU-4=95.8) and the highest win-rate (PWR=90.1%), consistent with style information being concentrated in late transformer layers.

6.4 Human Evaluation Study

To complement automated metrics, we conducted two light weight human evaluations using anonymized Forms: (i) an *Instructor* blind A/B preference + rubric study and (ii) a *Student* perception study. In the instructor study, participants reviewed programming prompts with a student attempt and compared two anonymous feedback responses (Feedback A vs. Feedback B), selecting an overall preference and rating multiple pedagogical criteria (e.g., technical correctness, authenticity, helpfulness, and trust-oriented tone) using a short rubric (Feedback A (very good), Feedback A (good), Neither (neutral), Both (equal), Feedback B (good), Feedback B (very good)). The form explicitly instructs instructors to judge pedagogical quality rather than minor grammatical issues and reports an estimated completion time of 8–12 minutes. In the student study, each participant reviewed

Table 4: Student survey average ratings (on a 1–5 Likert scale) across 20 respondents and 5 examples.

Statement	Rating (stdev)
The feedback was clear.	4.34 (1.05)
The feedback was helpful.	4.37 (1.04)
I trust this feedback.	4.31 (0.97)
It includes a full solution/code I could copy directly.	2.55 (1.49)
I know what the issue is after reading the feedback.	4.27 (1.07)
It sounds like a real instructor.	3.87 (1.34)

Table 5: Instructor survey across 12 respondents (blind A/B). Percentage of cases where feedback A (PERSA) was preferred over Feedback B (vanilla LLM); tie aggregates “Both”/“Neither” responses.

Statement	A (%)	B (%)	Tie (%)
Overall preference	83.6	1.8	14.6
Helpfulness / actionability	85.5	1.8	12.7
Authenticity (human instructor)	74.5	1.8	23.7
Trust-oriented tone	87.3	3.6	9.1
Technical correctness	61.8	5.5	32.7
Gives too much solution	27.3	27.3	45.4

five feedback instances and reported an evaluation of different features of the feedback on a Likert (1–5) scale.

Human-study interpretation. Tables 4 and 5 summarize our blind human evaluation comparing **PERSA** (Feedback A) to a vanilla LLM (Feedback B) across diverse respondents, including university instructors with varying teaching experience and students spanning beginner to advanced programming backgrounds. We used short, consistent rubrics that asked participants to judge clarity, helpfulness / action-ability, trust, and instructor-likeness (rather than minor grammar). We observe converging evidence that **PERSA**’s professor-style alignment improves perceived feedback quality beyond the baseline. Students report high ratings for clarity, helpfulness, trust, and understanding (4.27–4.37/5), while the lower score on “includes a full solution” (2.55) suggests the feedback typically supports revision without excessive solution dumping. Instructors similarly prefer **PERSA** (83.6%) and rate it higher on helpfulness/actionability (85.5%) and trust-oriented tone (87.3%), with substantially stronger perceived authenticity (74.5%) than the vanilla LLM.

To complement our automated evaluation, we conducted a human study involving 32 participants (comprising faculty and students) across 8 univer-

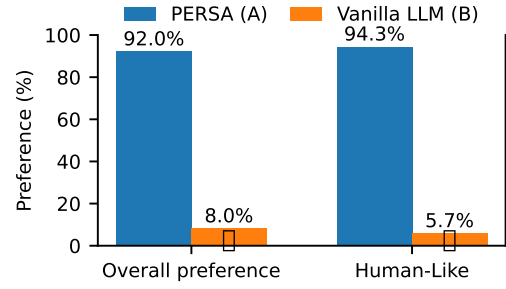


Figure 5: Human-study outcomes comparing **PERSA** against a vanilla LLM.

sities. This study compares **PERSA** against a vanilla LLM baseline to evaluate perceived pedagogical quality and alignment with instructor-like behavior. As shown in Figure 5, respondents consistently preferred **PERSA** over the vanilla baseline in both overall preference and human-like, instructor-style feedback, providing direct human-centered evidence that **PERSA** produces responses that are more authentic, supportive, and pedagogically aligned. Details of the study design are provided in Appendix C.

7 Conclusion

We presented **PERSA**, a professor-style reinforcement learning framework that adapts LLM-generated programming feedback to an instructor’s communicative voice while maintaining diagnostic correctness. **PERSA** follows an RLHF recipe, supervised fine-tuning, preference-based reward modeling, and PPO optimization, but makes adaptation practical and controllable by restricting learning to style-relevant parameters through *layer-selective*, parameter-efficient LoRA updates concentrated in the upper transformer layers. Across datasets and model backbones, **PERSA** achieves the strongest overall alignment, reaching near-ceiling style and lexical similarity (e.g., SAC/BLEU) without degrading correctness accuracy, and producing feedback that is consistently more structured, specific, and pedagogically actionable in qualitative analyses. Taken together, our results suggest that “instructor voice” behaves as a largely separable and optimizable attribute that can be injected via targeted adaptation rather than full-model retraining, enabling more trustworthy, authentic, and deployable AI feedback systems for education.

Limitations

PERSA is evaluated on a limited set of instructors, programming tasks, and feedback distributions, so its gains in stylistic alignment may not directly transfer to new courses, grading philosophies, or broader educational settings without additional preference data and recalibration. Moreover, because the reward model is learned from a specific instructor cohort and prompting setup, it may capture dataset-specific biases or stylistic artifacts, and, like RLHF more broadly, it remains vulnerable to reward misclassifications and over-optimization. Detailed discussion, limitations, and future directions are provided in Appendix D.

Ethical Considerations

We carefully considered the ethical responsibilities associated with this work, particularly the human evaluation component. The study was conducted in accordance with standard institutional and legal expectations for voluntary human participation, including anonymized response collection, minimal-risk survey design, and analysis in aggregate form without storing personally identifiable information.

References

- Bashaer Alsafari, Eric Atwell, Aisha Walker, and Martin Callaghan. 2024. Towards effective teaching assistants: From intent-based chatbots to llm-powered teaching assistants. *Natural Language Processing Journal*, 8:100101.
- Pronaya Bhattacharya, Vivek Kumar Prasad, Ashwin Verma, Deepak Gupta, Assadaporn Sapsomboon, Wattana Viriyasitavat, and Gaurav Dhiman. 2024. Demystifying chatgpt: An in-depth survey of openai’s robust large language models. *Archives of Computational Methods in Engineering*, 31(8):4557–4600.
- David Carless. 2012. Trust and its role in facilitating dialogic feedback. In *Feedback in higher and professional education*, pages 90–103. Routledge.
- Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. 2025. RLhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *ACM Computing Surveys*, 58(2):1–37.
- Dongwook Choi and Eunseok Lee. 2025. [Automated feedback generation for programming assignments through diversification](#). In *2025 IEEE/ACM 37th International Conference on Software Engineering Education and Training (CSEE&T)*, pages 230–241. IEEE.
- Wei Dai, Yi-Shan Tsai, Jionghao Lin, Ahmad Aldino, Hua Jin, Tongguang Li, Dragan Gašević, and Guanliang Chen. 2024. [Assessing the proficiency of large language models in automatic feedback generation: An evaluation study](#). *Computers and Education: Artificial Intelligence*, 7:100299.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259.
- Dorottya Demszky, Jing Liu, Heather C Hill, Dan Jurafsky, and Chris Piech. 2024. 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*, 46(3):483–505.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. RLhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Iria Estévez-Ayres, Patricia Callejo, Miguel Ángel Hombrados-Herrera, Carlos Alario-Hoyos, and Carlos Delgado Kloos. 2024. [Evaluation of LLM tools for feedback generation in a course on concurrent programming](#). *International Journal of Artificial Intelligence in Education*, 35(2).
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.
- Richard M Felder, Linda K Silverman, and 1 others. 1988. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Hagit Gabbay and Anat Cohen. 2024. [Combining LLM-generated and test-based feedback in a MOOC for programming](#). In *Proceedings of the 11th ACM Conference on Learning @ Scale (L@S 2024)*, pages 177–187. ACM.

- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5484–5495. Association for Computational Linguistics.
- Luke Gusukuma, Austin Cory Bart, and Dennis Kafura. 2020. [Pedal: An infrastructure for automated feedback systems](#). In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. ACM.
- Jieun Han, Haneul Yoo, Junho Myung, Minsun Kim, Hyunseung Lim, Yoonsu Kim, Tak Yeon Lee, Hwajung Hong, Juho Kim, So-Yeon Ahn, and 1 others. 2023. Llm-as-a-tutor in efl writing education: Focusing on evaluation of student-llm interaction. *arXiv preprint arXiv:2310.05191*.
- Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and 1 others. 2021. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.
- Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.
- Uday Kamath, Kevin Keenan, Garrett Somers, and Sarah Sorenson. 2024. Tuning for llm alignment. In *Large Language Models: A Deep Dive: Bridging Theory and Practice*, pages 177–218. Springer.
- Hieke Keuning, Johan T. Jeuring, and Bastiaan Heeren. 2018. [A systematic literature review of automated feedback generation for programming exercises](#). *ACM Transactions on Computing Education*, 19(1):1–43.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*.
- Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2022. [Neuron-level interpretation of deep nlp models: A survey](#). *Transactions of the Association for Computational Linguistics*, 10:1285–1303. Surveys methods for interpreting style-related signals in transformer models, relevant for identifying style-specific neurons in higher layers.
- Wen Lai, Viktor Hangya, and Alexander Fraser. 2024. Style-specific neurons for steering llms in text style transfer. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13427–13443.
- Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, and 1 others. 2024. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196.
- Hong-Yi Lin, Chunhua Liu, Haoyu Gao, Patanamon Thongtanunam, and Christoph Treude. 2025. Codereviewqa: The code review comprehension assessment for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9138–9166.
- Euan D Lindsay, Mike Zhang, Aditya Johri, and Johannes Bjerva. 2025. The responsible development of automated student feedback with generative ai. In *2025 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–10. IEEE.
- John Loughran. 2019. Pedagogical reasoning: The foundation of the professional knowledge of teaching. *Teachers and Teaching*, 25(5):523–535.
- Yun Luo, Zhen Yang, Fandong Meng, Yuchen Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.
- Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*.
- Milad Moradi, Ke Yan, David Colwell, Matthias Samwald, and Rhona Asgari. 2024. Exploring the landscape of large language models: Foundations, techniques, and challenges.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Sinan Ozdemir. 2023. *Quick start guide to large language models: strategies and best practices for using ChatGPT and other LLMs*. Addison-Wesley Professional.
- 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.
- Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*.

- Tung Phung, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. 2023. Generating high-precision feedback for programming syntax errors using large language models. *arXiv preprint arXiv:2302.04662*.
- Mahefa Abel Razafinirina, William Germain Dimbisoa, and Thomas Mahatody. 2024. Pedagogical alignment of large language models (LLM) for personalized learning: a survey, trends and challenges. *Journal of Intelligent Learning Systems and Applications*, 16(4):448–480.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Tianhao Shen, Chenyang Zhou, Tianyu Wu, Canwen Wang, Xuming Wang, and Steven C. H. Hoi. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*. Discusses challenges in evaluating LLMs, including metrics for content correctness and alignment, relevant for Correctness Accuracy (CA) in feedback evaluation.
- Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, and Xiao-Ming Wu. 2024. Understanding layer significance in llm alignment. *arXiv preprint arXiv:2410.17875*.
- Hakim Sidahmed, Samrat Phatale, Alex Hutcheson, Zhuonan Lin, Zhang Chen, Zac Yu, Jarvis Jin, Simral Chaudhary, Roman Komarytsia, Christiane Ahlheim, and 1 others. 2024. Parameter efficient reinforcement learning from human feedback. *arXiv preprint arXiv:2403.10704*.
- Sinan Sonlu, Bennie Bendiksen, Funda Durupinar, and Uğur Güdükbay. 2024. The effects of embodiment and personality expression on learning in llm-based educational agents. *arXiv preprint arXiv:2407.10993*.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, and 1 others. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.
- Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.
- Di Wu, Meng Chen, Xu Chen, and Xing Liu. 2024a. Analyzing k-12 ai education: A large language model study of classroom instruction on learning theories, pedagogy, tools, and ai literacy. *Computers and Education: Artificial Intelligence*, 7:100295.
- Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024b. Dpo: Direct preference optimization with dynamic. *Advances in Neural Information Processing Systems*, 37:129944–129966.
- Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwa, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, and 1 others. 2024. Sorry-bench: Systematically evaluating large language model safety refusal. *arXiv preprint arXiv:2406.14598*.
- Yuzi Yan, Xingzhou Lou, Jialian Li, Yiping Zhang, Jian Xie, Chao Yu, Yu Wang, Dong Yan, and Yuan Shen. 2024. Reward-robust rlhf in llms. *arXiv preprint arXiv:2409.15360*.
- Zhe Victor Zhang and Ken Hyland. 2022. Fostering student engagement with feedback: An integrated approach. *Assessing Writing*, 51:100586.

Appendix

A Pseudo Code

Algorithm 1: PERSA: Layer-Selective LoRA + RLHF

Input : Base LLM π_θ ; demos $\mathcal{D}_{\text{SFT}}$; prefs $\mathcal{D}_{\text{pref}}$; top layers L ; LoRA rank r ; KL β ; PPO steps T

Output : Style-aligned policy $\pi_{\theta'}$

Freeze θ ; insert LoRA (rank r) in top L layers; trainable params θ_{LoRA} ;

SFT;;

$\theta_{\text{LoRA}} \leftarrow \arg \min_{\theta_{\text{LoRA}}} \mathbb{E}_{(x, y^*) \sim \mathcal{D}_{\text{SFT}}} [-\sum_t \log \pi_\theta(y_t^* | x, y_{<t}^*)]$;

Set reference $\pi_{\text{ref}} \leftarrow \pi_\theta$ (frozen);

RM;;

Train reward r_ϕ on $\mathcal{D}_{\text{pref}}$ with

$\mathcal{L}_{\text{RM}} = -\mathbb{E} [\log \sigma(r_\phi(x, y^{(w)}) - r_\phi(x, y^{(l)}))]$;

PPO;;

for $t = 1$ **to** T **do**

Sample x ; generate $y \sim \pi_\theta(\cdot | x)$; compute

$\tilde{R} = r_\phi(x, y) - \beta \text{KL}(\pi_\theta \| \pi_{\text{ref}})$;

Update θ_{LoRA} by PPO using reward $\tilde{R}$ (clipped ratio);

return $\pi_{\theta'}$ (base + trained LoRA);

B Detailed Experimental Settings

B.1 Data Splits & Leakage Controls.

To prevent train-test contamination and inflated SAC/BLEU scores from near-duplicate examples, we employ leakage-aware splits and explicit overlap audits. In addition to a standard *instance-level* split, we evaluate on a stricter *New-Problems* split that withholds entire problem IDs from training. Before splitting, we remove exact duplicates using hashes over {problem, solution, reference feedback} and filter near-duplicates via Min-Hash/Jaccard similarity on problem text and reference feedback. After splitting, we compute each test example’s maximum similarity to the training set and verify that all remain below a conservative threshold. To further reduce dependence on surface-level overlap, we also report paraphrase-robust SAC and diversity statistics. Table 6 summarizes these controls.

Preference data. The same three datasets are used to construct both $\mathcal{D}_{\text{SFT}}$ and $\mathcal{D}_{\text{pref}}$. For $\mathcal{D}_{\text{SFT}}$, each example is a prompt-reference pair (x, y^*) , where y^* is the professor or expert-written feedback. For $\mathcal{D}_{\text{pref}}$, we form pairwise comparisons $(x, y^{(w)}, y^{(l)})$ using the same training prompts: the preferred response $y^{(w)}$ is the professor/expert feedback, and the rejected response $y^{(l)}$ is feedback gen-

Table 6: Leakage-aware splits & overlap audits.

Audit / Split		Std. New-Problems	
Split key		inst.	prob. ID
Exact dups rm. (%) ↓		1.2	1.2
Near-dups rm. (%) ↓		6.5	6.5
Max train-test (problem) ↓	sim.	0.23	0.19
Max train-test (ref. fb) ↓	sim.	0.18	0.15
# test items >0.8 (Jaccard) ↓		0	0
Paraphrase SAC (%) ↑	agr.	94.0	91.5
BERTScore (F1) ↑		0.92	0.90
Self-BLEU ↓		0.54	0.50
Distinct-2 ↑		0.21	0.24

Table 7: Method footprint. PERSA retains an RLHF pipeline but updates only Top- L LoRA adapters (layer-selective PEFT), improving efficiency and style fidelity.

Method	Obj.	RM	On-pol.	Trainable	Scope	Pros
Base	–	N	N	0%	Frozen	No cost
SFT	SFT	N	N	Full	All	Stable, simple
RLHF	PPO	Y	Y	Full	All	Strong align; costly
DPO	DPO	N	N	Full	All	Offline, stable
ORPO	ORPO	N	N	Full	All	Efficient prefs
KTO	KTO	N	N	Full	All	Robust prefs
PERSA	SFT+PPO	Y	Y	Top- L LoRA	Top- L	Style + low drift

erated by the vanilla backbone or a weaker baseline under the same prompt. Preference pairs are created only from the training split to avoid train-test leakage.

B.2 Model Evaluation Framework

We use lightweight, open-weight instruction-tuned LLMs as backbones so that PERSA (SFT + reward modeling + PPO) can run in a single-GPU setting. **Vanilla LLM baseline.** Unless otherwise stated, “vanilla LLM” refers to the instruction-tuned backbone used without task-specific fine-tuning or preference optimization. In our experiments, this corresponds to google/gemma-2-2b-it or meta-llama/Llama-3.2-3B-Instruct, evaluated with the same prompts and decoding configuration as the adapted models.

Primary backbone: Gemma 2 2B Instruct. We use google /gemma-2-2b-it as the main policy model due to its strong instruction-following behavior at a practical size, enabling parameter-efficient RLHF in constrained compute (Team et al., 2024). **Secondary backbone: Llama 3.2 3B Instruct.** We additionally evaluate meta-llama/Llama-3.2-3B-Instruct to test

cross-family robustness (Meta vs. Google) while keeping the model small enough for Colab-scale tuning (Dubey et al., 2024). **General recipe (model-agnostic).** All experiments treat the LLM as a conditional policy $\pi_\theta(y \mid x)$ over feedback tokens; we apply (1) supervised fine-tuning on instructor demonstrations, (2) reward modeling from pairwise preferences, and (3) PPO-based policy optimization with a KL constraint to preserve base knowledge. This structure is compatible with any decoder-only LLM that supports PEFT adapters (Ouyang et al., 2022; Schulman et al., 2017).

Methods Compared: We compare PERSA against strong baselines spanning supervised imitation and preference-based alignment to assess their ability to generate professor-like programming feedback while preserving correctness. (1) **Base Model** denotes the instruction-tuned backbone (Gemma-2-2B-IT or Llama-3.2-3B-Instruct) used *without* any task-specific adaptation. (2) **Supervised Fine-Tuning (SFT)** fine-tunes the backbone on professor-authored demonstrations using a maximum-likelihood objective. (3) **InstructGPT-style RLHF** follows the standard RLHF recipe of reward modeling from preferences and policy optimization with PPO and KL control (Ouyang et al., 2022). To reflect recent state-of-the-art alternatives to PPO-based RLHF, we additionally include three *offline* preference optimization methods trained on the same professor preference pairs: (4) **DPO** (Direct Preference Optimization), which directly optimizes the policy from preference comparisons without an explicit reward model (Wu et al., 2024b), (5) **ORPO** (Odds Ratio Preference Optimization), which integrates preference signals into an SFT-like objective (Hong et al., 2024), and (6) **KTO** (Kahneman–Tversky Optimization), a robust preference based objective inspired by prospect theory (Ethayarajh et al., 2024). Finally, (7) **PERSA** is our proposed framework that combines SFT initialization with *layer-selective, parameter-efficient* preference alignment (LoRA on upper layers) to target instructor style while minimizing drift in core capabilities. **Table 7 columns.** We report each method’s optimization objective (*Obj.*), whether it requires an explicit reward model (*RM*) and on-policy rollouts (*On-pol.*), the fraction of trainable parameters (*Trainable*), and update scope across layers (*Scope*), highlighting that offline preference methods (DPO/ORPO/KTO) train full weights without RM/rollouts, while **PERSA** follows an RLHF pipeline (RM+on-policy) but updates only Top- L

LoRA adapters for efficient, style-focused alignment. All approaches are evaluated under the same prompts, data splits, and decoding settings, enabling a controlled comparison of stylistic alignment and content fidelity.

B.3 Evaluation Metric

We evaluate PERSA on two complementary dimensions: *style alignment* (how feedback is phrased) and *diagnostic fidelity* (whether feedback is correct). All metrics are reported on the held-out test set as means.

Style Alignment Score (SAC). We train a calibrated binary style classifier (professor vs. non-professor feedback) and compute

$$\text{SAC} = \frac{1}{N} \sum_{i=1}^N C(\hat{y}_i), \quad (8)$$

where $C(\hat{y}_i) \in [0, 1]$ is the posterior for the professor label; higher is better (Fu et al., 2018).

Average Politeness Closeness (APC). Using a politeness scorer $p(\cdot) \in [0, 1]$ (Danescu-Niculescu-Mizil et al., 2013), we measure tone closeness to the professor reference:

$$\text{APC} = \frac{1}{N} \sum_{i=1}^N \left(1 - |p(\hat{y}_i) - p(y_i^*)|\right). \quad (9)$$

BLEU-4. We report corpus BLEU-4 (with smoothing) to quantify n -gram overlap in phrasing with the professor feedback (Papineni et al., 2002).

Correctness Accuracy (CA). CA measures whether the feedback correctly identifies solution correctness. Let z_i be the unit-test label and $\hat{z}_i$ the judgment extracted from $\hat{y}_i$; then

$$\text{CA} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\hat{z}_i = z_i], \quad (10)$$

with higher indicating more reliable feedback (Shen et al., 2023). We interpret CA as a preservation check rather than the main source of improvement.

Preference Win Rate (PWR). To reflect RLHF objectives, we report the fraction of prompts where PERSA is preferred over a baseline under the same rater:

$$\text{PWR} = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left[r_\phi(x_i, \hat{y}_i^{\text{PERSA}}) > r_\phi(x_i, \hat{y}_i^{\text{BASE}}) \right]. \quad (11)$$

a standard RLHF evaluation signal (Ouyang et al., 2022).

Metric reliability. We treat SAC and APC as automatic proxies rather than definitive measures of pedagogical quality. To check reliability, we calibrate the SAC classifier on held-out professor/non-professor feedback. For APC, we report correlation with human judgments of tone/trust-oriented feedback. These checks are intended to ensure that automatic metrics track human-perceived style dimensions rather than only surface lexical overlap.

B.4 Configurations

We summarize key hyperparameter for reproducible local-scale training; values below are defaults and can be scaled with model size.

Input formatting. We serialize each example as: *Problem + Student code + optional tests/constraints* $\rightarrow$ *Feedback*. We cap the input length (e.g., 1 2k tokens) and truncate from the left (oldest context) when needed.

Parameter-efficient tuning (LoRA/QLoRA). We inject LoRA adapters into attention and MLP projections (e.g., `q_proj`, `k_proj`, `v_proj`, `o_proj`, `up_proj`, `down_proj`, `gate_proj`) and update only the *top L transformer blocks* (layer-selective adaptation). Typical settings: rank $r \in \{8, 16\}$, $\alpha \in \{16, 32\}$, dropout $p = 0.05$. For coding environments, we use 4-bit NF4 quantization with double-quantization when available (QLoRA). (Hu et al., 2022; Detrmers et al., 2023)

SFT. Optimizer: AdamW; learning rate $1e-5$ $2e-5$; epochs 3 10 (smaller for CodeReviewQA, larger for the 200-example instructor set); effective batch size via gradient accumulation; warmup 3 5%; weight decay 0.0 0.1; label smoothing optional (0.0 0.1).

Reward model (RM). We train a reward model $r_\phi(x, y)$ on preference pairs with a Bradley Terry / logistic loss; batch size 8 32; 1 3 epochs; early stopping on validation preference accuracy (Ouyang et al., 2022).

PPO (RLHF). We optimize the KL-regularized objective with PPO using: rollout batch 32 128 prompts, 1-4 PPO epochs per iteration, clipping $\epsilon = 0.2$, KL coefficient $\beta \in [0.005, 0.05]$, and a reference policy π_{ref} set to the post-SFT model. We keep generation short for feedback (e.g., max new tokens 128 256) to stabilize advantage estimates. (Schulman et al., 2017; Ouyang et al., 2022)

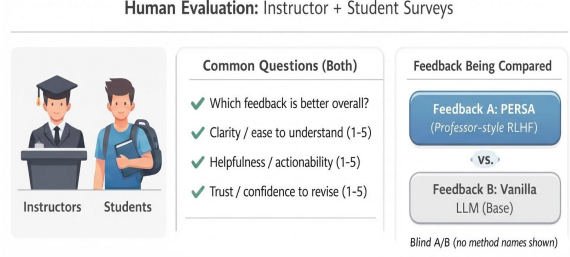


Figure 6: Overview of the human evaluation design used for instructor and student studies.

Compute and reproducibility. All runs fix random seeds, log training curves, and report mean $\pm$ std over 3 seeds when feasible. On GPUs (A100), Gemma-2-2B with 4-bit LoRA typically fits in memory with moderate batch sizes; Llama-3.2-3B may require smaller rollouts or more accumulation.

C Human Study Design

Figure 6 summarizes our human evaluation protocol, showing the participating groups (instructors and students), the shared evaluation criteria (overall preference, clarity, helpfulness, and trust), and the blind A/B comparison between PERSA-generated feedback and vanilla LLM feedback used in both surveys.

Our instructor student **human evaluation section C**, provides direct evidence that PERSA’s RLHF objective translates into perceived pedagogical gains participants overwhelmingly preferred the PERSA outputs over a vanilla LLM thereby validating that the learned preference signal improves feedback quality beyond automated style metrics. **Figure 5** states that across instructor and student respondents, PERSA is preferred over the vanilla LLM (92% vs. 8%) and is judged to better match a human-instructor feedback style (94.3%), providing human-centered validation of PERSA’s RLHF-based style alignment.

D Discussion

PERSA demonstrates that *personalized pedagogical alignment* can be achieved by tuning *how* feedback is delivered (tone, structure, and phrasing) without degrading *what* is delivered (diagnostic correctness). Across three datasets and two model families, we observe a consistent separation between *style* and *substance*: base instruction-tuned models retain high correctness (CA) but exhibit weak professor-style fidelity (low SAC/BLEU),

while preference-based methods systematically improve stylistic alignment with minimal impact on CA. Crucially, PERSA achieves the strongest overall alignment (near-ceiling SAC/BLEU with $CA \approx 100\%$), indicating that instructor voice is an optimizable target distinct from generic “helpfulness” alignment.

Why each component matters. PERSA’s performance is not due to a single stage, but to the complementary roles of its components. **SFT** provides a stable initialization that learns the instructor’s canonical feedback template (e.g., encouragement → diagnosis → fix → verification), yielding the largest first jump in SAC/BLEU and perfect CA, but it does not fully capture nuanced preferences such as strictness, specificity, and emphasis on edge cases. **Reward modeling** operationalizes these nuanced stylistic and pedagogical preferences from pairwise comparisons, producing a scalar signal that encodes instructor-specific judgments beyond likelihood matching. **PPO with KL control** then refines the policy to maximize preference reward while preventing broad drift from the SFT reference, which is essential for preserving the backbone’s general coding competence and avoiding capability regressions.

Why layer-selective LoRA is central. A key insight behind PERSA is that updating *all* parameters is neither necessary nor desirable for style personalization. Motivated by evidence that late transformer layers and FFN submodules disproportionately encode high-level attributes (discourse patterns, tone, and stylistic signatures), PERSA restricts trainable parameters to LoRA adapters in the top layers. This yields two benefits: (i) **efficiency** only a small fraction of parameters (e.g., $\sim 30M$ in the last 4 layers) need to move to achieve large stylistic gains; and (ii) **stability** constraining updates reduces catastrophic drift and preserves correctness, which is critical in educational feedback where factual errors can harm learning. The ablations support this design: PPO without SFT is sample-inefficient for learning instructor voice, and broader updates (full-parameter PPO or all-layer LoRA) improve style but underperform targeted top-layer adaptation, implying that “style-bearing capacity” is concentrated in late layers rather than uniformly distributed.

Relevance and broader implications. PERSA’s results suggest a practical path to **course-level personalization** of AI tutors: rather than training new models per instructor, one can adapt a compact

backbone with a small set of instructor-specific adapters and a preference signal, enabling rapid deployment and easy swapping of styles across courses. This is particularly relevant for educational settings where instructors have distinct grading philosophies (e.g., emphasis on readability vs. correctness vs. edge cases) and where student trust depends on perceived authenticity. Beyond education, the same principle generalizes to domains requiring **persona consistent communication** (e.g., customer support, clinical explanations, legal drafting) where maintaining core competence while matching a specific communication style is essential. Finally, the strong agreement between quantitative alignment metrics and qualitative examples indicates that optimizing for instructor preference can yield feedback that is not only stylistically similar but also more actionable and pedagogically structured, supporting the use of preference based objectives as a reliable mechanism for fine-grained style alignment.

What PERSA adds beyond standard RLHF. PERSA does not claim a new reinforcement learning algorithm; rather, it operationalizes RLHF for a narrower educational alignment problem: instructor-specific feedback style. Its novelty lies in treating professor voice as a distinct optimizable attribute, combining instructor-authored demonstrations, preference modeling, and layer-selective LoRA updates to shift feedback tone, structure, and specificity while limiting parameter drift. This framing differs from generic RLHF, which optimizes broad helpfulness, and from standard SFT, which imitates demonstrations without explicitly optimizing instructor-preference signals.

Limitations. Our evaluation covers a small number of instructors, student feedback distributions, and programming tasks; consequently, the observed style improvements may not directly transfer to new instructors, grading policies, courses, or broader educational settings without additional preference data and validation. In addition, because PERSA is trained with a fixed prompting/template and a specific instructor cohort, the reward model may capture prompt- or dataset-specific artifacts, which can reduce robustness when moving to different assignment formats, programming languages, or institutional contexts unless recalibrated. Finally, as with RLHF more broadly, a learned reward can reflect annotator bias and remains vulnerable to misspecification or over-optimization even with KL regularization, motivating careful monitoring

and human-in-the-loop oversight prior to classroom deployment (Ouyang et al., 2022; Lindsay et al., 2025).

While our human study partially validates the automatic style metrics, we do not yet include an LLM-as-a-judge evaluation for holistic pedagogical style. Future work will compare SAC/APC with expert-calibrated LLM judges that assess authenticity, encouragement, specificity, and instructional tone.

Future Work. We will extend PERSA to multi-instructor and multi-course settings with longitudinal classroom studies, and will explore stronger preference collection and safety controls (e.g., uncertainty-aware rewards and calibration) to ensure style-personalized feedback remains accurate, fair, and robust in real deployments.

Educational outcomes beyond style. to complement style/correctness metrics (SAC/APC/BLEU/CA/PWR), we will evaluate *learning-relevant* outcomes: (i) revision success by measuring whether students can fix the identified issues on a resubmission task (pass@1 / unit-test pass rate improvement), and (ii) error recurrence by tracking whether the same misconception/bug type reappears in a follow-up attempt. In addition, we will assess **action-ability and rubric alignment** using a validated feedback rubric, rated by instructors/TA annotators, rather than preference alone.

Data-lean fine-tuning of models for evaluating teacher performance in a GenAI-led elicitation simulation

Beata Beigman Klebanov Andrew Hoang Jamie Mikeska Benny Longwill
Sanjna Kashyap Shreyashi Halder Aakanksha Bhatia*

ETS Research Institute, Princeton, NJ, USA

{bbeigmanklebanov, ahoang, jmikeska, blongwill, skashyap, shalder001}@ets.org

aakankshabhatia01@gmail.com

Abstract

Recent advances in the capabilities of conversational agents based on large language models make them a very promising tool for role playing K-12 students in order to train educators in conversational teaching practices, such as eliciting student thinking, explaining disciplinary content, and facilitating a classroom discussion. In fact, such simulations can and have been developed relatively quickly and without data to machine-learn from – neither classroom data nor human-simulated data. To enhance the usefulness and effectiveness of such teaching simulations, it is necessary to provide pedagogically sound, timely, and personalized feedback to the educator about their simulation performance. In this study, we present experiments on fine-tuning models to evaluate educator performance in an elicitation teaching simulation. The models are developed with data collected during usability testing of the simulation and evaluated on real user data. We show that even with relatively little fine-tuning data, robust performance can be obtained.

1 Introduction

Digital teaching simulations (**DTS**) have increasingly been integrated into teacher education settings to support educators in learning how to engage in high-leverage teaching practices, such as eliciting student thinking, facilitating productive discussions, and communicating with families, and have been shown to improve educators' skills, knowledge and beliefs (Dalinger et al., 2020; Dieker et al., 2019; Ersozlu et al., 2021; Lee et al., 2024; Mikeska et al., 2025a). One of the most common types of DTS involves educators in conversing with one or more student avatars via an online interface where the avatars appear visually and talk to each other and the educator in real time. In these simulations, there is a human-in-the-loop called a

simulation specialist who has been trained to use specialized technology, such as voice modulation software and game controllers, and to enact various student personas that align with the student avatars' pre-specified personalities, backgrounds, previous experiences, and content understandings (Bondie et al., 2021).

Research has shown that productive simulation-based learning experiences involve instructional or learning cycles whereby teacher educators or professional development facilitators support educators in preparing for, engaging in, and then debriefing and reflecting on their simulation performances (Lottero-Perdue et al., 2023; Mikeska et al., 2023a; Woods et al., 2024). These instructional cycles can also include formative feedback from a teacher educator, coach, educators' peers, or via written feedback reports. Research has suggested that such feedback supports educators in identifying their instructional strengths and areas for growth and in learning how they can apply what they are learning to K-12 classrooms (Cohen et al., 2020; Horn et al., 2023; Mikeska et al., 2023b; Pecore et al., 2023). However, despite the empirical evidence for the efficacy of DTS, they are challenging to scale due to the need for the simulation specialist to power the student avatars during each simulation.

To address this scaling challenge, recent research has begun exploring how generative AI (**GenAI**) can be used to simulate student responses (Marquez-Carpintero et al., 2025; Hong et al., 2025; Liu et al., 2026; Pan et al., 2025; Markel et al., 2023; Lim et al., 2025; Zhang et al., 2026; Zhuang and Zhang, 2025; Lee et al., 2025, 2026; Gíslason, 2026; Ziying et al., 2026; Son, 2025). While users perceive GenAI teaching simulations as useful to support their learning (Lim et al., 2025; Markel et al., 2023; Mikeska et al., 2025b; Zhang et al., 2026) and manual analyses are being applied to explore user activity in a simulation (Lee et al., 2026; Son, 2025), we are not aware of studies focusing

*The author was at ETS at the time of research.

on automated assessment of teacher performance in a GenAI simulation for formative purposes.

Prior work on automated evaluation of teacher performance has been conducted on annotated classroom discussions data, such as mathematics discussions (TalkMoves (Suresh et al., 2022), NCTE (Demszky and Hill, 2023)), ELA discussions (Tran et al., 2024), small-group discussions in DTS (Asano et al., 2025; Nazaretsky et al., 2023), and tutoring dialogues (Cao et al., 2025; Macina et al., 2023). In contrast with at least a hundred teacher performances in each of the above, we explore feasibility of evaluating teacher performance in a newly created GenAI-led simulation with very little development data – only 40 performances, collected during usability testing of the simulation.

2 Related work

A variety of constructs related to effective teaching have been modeled computationally based on human-annotated datasets, predominantly of real or simulated classroom discussions and tutorial dialogues. These include specific teaching actions such as asking effective questions (Alic et al., 2022; Jensen et al., 2020; Kelly et al., 2018), encouraging students' interaction (Ilagan et al., 2024), supporting student motivation (Hunkins et al., 2022), enacting conversational uptake (Demszky et al., 2021; Jensen et al., 2020); more comprehensive analyses of various teacher and student "talk moves", common types of teacher and student contributions to classroom discourse (Suresh et al., 2022; Tran et al., 2024; Cao et al., 2025; Wang et al., 2025), as well as holistic characteristics of discussions such as coherence (Asano et al., 2025), extent of instructional support (Whitehill and LoCasale-Crouch, 2024), and instructional quality (Xu et al., 2024).

Methods used for automated analyses of educational discourse ranged from feature-based machine learning (Donnelly et al., 2017; Jensen et al., 2020; Kelly et al., 2018; Hunkins et al., 2022; Samei et al., 2014), to deep neural networks, including transformers (Demszky et al., 2021; Schlotterbeck et al., 2021; Kupor et al., 2023; Nazaretsky et al., 2023; Alic et al., 2022; Xu et al., 2024; Ilagan et al., 2024), to large language models (LLMs) using techniques such as few-shot prompting, chain-of-thought, and parameter-efficient fine-tuning (Wang and Demszky, 2023; Kupor et al., 2023; Whitehill and LoCasale-Crouch, 2024; Hou et al., 2024; Wang et al., 2025; Cao et al., 2025).

Wang et al. (2025) compared LLM prompt-engineering, full fine-tuning with BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), and parameter-efficient fine-tuning (PEFT) of LLMs on the TalkMoves dataset. They found that PEFT approaches were stronger on the talk moves Restating and Revoicing of the students' utterances but performed comparably to RoBERTa on the other constructs. However, these findings were obtained with fine-tuning on hundreds of transcripts totaling about 1.5M teacher words. Beigman Klebanov et al. (2025) studied few-shot vs fine-tuned models on about 1K teacher utterances in small-group discussion DTS and found fine-tuned models to be superior; however, the annotated data included only select, clear-cut utterances per transcript. Tran et al. (2024) evaluated both fine-tuned BERT and few-shot LLMs on detecting productive teaching moves and found that fine-tuned models were generally competitive though yielded to LLMs in certain configurations. Focusing specifically on the amount of fine-tuning data in the broader context of text classification, Zhang et al. (2025) compared BERT-like fine-tuned models, zero-shot LLMs, and LLM-internal-states-based methods on six text classification tasks and concluded that BERT-like models were stronger for what the authors termed "general text classification," but the strong performance of BERTs was predicated on tuning on moderate to abundant data. When data size was reduced to a few hundred (up to 450 instances, comparable to our scenario), they sank below GPT-4o results for the three datasets on which both model types of evaluated. In the current study, we investigate a data-lean scenario focusing on fine-tuned models, with an LLM-based few-shot model as a baseline.

3 The GenAI-led teaching simulation

The simulation was designed to support elementary pre-service and in-service teachers and paraeducators in developing the teaching competency of eliciting student thinking in elementary mathematics. During the simulation, teachers engaged in voice conversation with the Cecilia chatbot. Cecilia's responses were generated in accordance with a predefined student profile (section 3.1), allowing teachers to practice instructional strategies to elicit Cecilia's method for solving the addition word problem and her understandings and misunderstandings around addition. Cecilia was designed to respond to the teacher's questions with

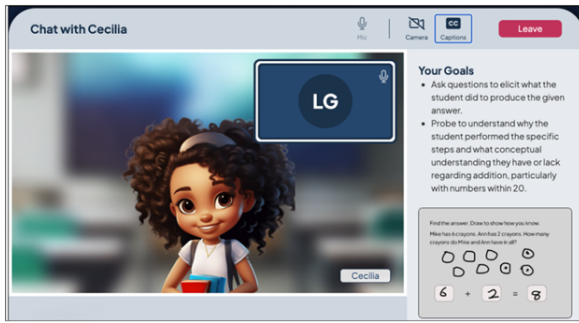


Figure 1: Interface of the GenAI teaching simulation.

the minimal amount of information that was responsive to the question, thus allowing the teacher to practice elicitation. That is, when asked a question like ‘What did you do to solve the problem?’, Cecilia would describe the first step but not the full process and would not divulge her reasoning unless explicitly prompted to do so by the teacher.

3.1 The strategies for adding task

The Strategies for Adding task is based on design criteria for elicitation tasks (Qi and Sykes, 2016). Cecilia’s knowledge profile specifies both her understanding and gaps in knowledge when solving addition problems. In this task, Cecilia solved the word problem: “Mike has 6 crayons. Ann has 2 crayons. How many crayons do Mike and Ann have in all?” Cecilia solved this problem and showed her understanding in her written work (Figure 1 bottom right). She understands that 6 represents Mike’s crayons, 2 represents Ann’s crayons, and 8 represents the total. Cecilia counts on from the first addend and uses models (e.g., drawings) to represent and solve the problem. However, she cannot add numbers fluently. She does not understand that counting on from the larger addend (regardless of the position) is more efficient. She is not familiar with any strategies besides drawing and counting for addition and she does not understand the commutative property. This knowledge profile represents a step in a typical progression of learning addition for early elementary students.

In the simulation, teachers were not provided with detailed information about Cecilia’s conceptual understanding. They were instructed to elicit what Cecilia did and did not understand regarding addition (top right – ‘Your Goals’ – in Figure 1). A 10-minute preparation time was provided, allowing teachers to consider their elicitation strategies.

3.2 Implementation of the simulation

Chatbot default dialogue responses were generated using Azure OpenAI’s GPT-4o model (v2024-08-06) via Microsoft Azure’s Chat Completions API. Each request was issued as a single-turn completion conditioned on a system prompt that encoded the knowledge profile of the simulated student, instructional constraints, and worked examples along with few-shot dialog examples and the full conversation history.

Response generation followed a hybrid pipeline combining LLM-based dialogue generation with deterministic validation and filtering. Mathematical expressions in user input were normalized using GPT-4o, then parsed and evaluated using Python’s Abstract Syntax Tree (AST) library to enforce structural constraints on practice problems (i.e., a single addition operation with two operands and a sum of up to 20). Inputs that failed the structural requirements triggered replacement of the initial response with predefined messages (e.g., “Oops! I don’t understand this!”).

In addition to structural validation on user input, regular expression-based checks were applied to detect mathematical vocabulary that Cecilia should not be familiar with (e.g., equation) and out-of-range numerical values (i.e., any number greater than 20), with violations resulting in appropriate canned responses (e.g., “I don’t know what that means” and “That’s too many to draw”, respectively). If no constraints were violated, a final LLM-based simplification step adjusted lexical complexity and limited responses to a single procedural step. A more comprehensive description of the implementation of the chatbot and the evaluation of the accuracy of Cecilia’s responses in terms of responsiveness to the teacher’s questions; appropriateness in age, emotional tone, amount of information provided, and coherence with respect to prior conversation; and alignment with the student’s knowledge profile can be found in Mikeska et al. (2025b).

4 Data

4.1 Participants

Study 1: A total of 45 educators (paraeducators, in-service teachers, and principals) participated in a usability study to test the simulation. The age range was 25 to 64 years. Among them, 67% identified as White, 18% as Hispanic or Latino, 13% as Black or African American, and 4% as Asian or Asian American. The majority of educators were

female (82%). Regarding educational background, 67% held a Master's degree, 13% a Bachelor's degree, 11% a Doctoral degree, 7% other degrees, and 2% a high school diploma. Additionally, 76% of the educators had prior experience in teaching mathematics. Their familiarity with digital teaching simulations varied: 11% of were very familiar, 31% were somewhat familiar, 38% were a little familiar, and 20% were not familiar. We collected data from 45 participants; however, due to technical issues, we had to exclude 5 transcripts from the analysis. The 40 transcripts collected in this study comprised the **development** dataset.

Study 2: This study was the first implementation pilot of the simulation within a teacher education context. The data was collected from 10 pre-service teachers (PSTs) enrolled in a mathematics method course at a private college in the Northeastern USA. PSTs were aged 18-24. At the time of data collection, 90% of them had obtained a high school diploma. Half of the participants had experience as substitute teachers or in other teaching capacities. In terms of familiarity with digital teaching simulations, 50% were not familiar, 30% were somewhat familiar, and 20% were a little familiar. The gender and ethnicity data for PSTs were not collected.

In this study, as part of the coursework, PSTs participated in two instructional cycles using the Strategies for Adding simulation. During the first cycle, PSTs prepared for the simulation by reviewing the simulation's goal and the scenario's background. The 10-minute preparation phase included reviewing Cecilia's written work and considering questions or prompts to use to elicit her problem-solving process and conceptual understanding of addition. Next, they engaged in the simulation, followed by a reflection session during their 50-minute class time. Their mathematics methods teacher educator facilitated the reflection activity, during which PSTs reviewed [Hallman-Thrasher et al. \(2020\)](#) categorization of questions by purpose and analyzed their own transcript. Each PST then compared their transcript with a partner's to further reflect on how to elicit Cecilia's thinking and modify their questioning strategy in the future.

After one week, the second instructional cycle took place. After the second simulation, PSTs engaged in a reflection activity and categorized their questions again. In this reflection, they focused on the information each question gathered about Cecilia's mathematical understanding. They participated in a whole-class discussion where PSTs

analyzed Cecilia's conceptual understanding and knowledge gaps based on their simulation experiences. Afterward, they completed a debrief assignment to reflect on their elicitation techniques. The 20 transcripts collected in this study comprised the **test** dataset.

4.2 Annotation of aspects of teaching quality

Facilitating productive talk in classrooms with and between students requires that teachers use specific instructional moves to gauge students' initial and developing ideas and prompt them to share their thinking and understanding ([Michaels and O'Connor, 2015](#); [Wei et al., 2018](#)). Eliciting student thinking requires teachers to attend to specific student ideas, ask follow-up questions, and explain their reasoning ([Estapa and Davis, 2023](#); [Shaughnessy and Boerst, 2018](#)). In this study, we focus on three key constructs related to the practice of eliciting student thinking summarized in Table 1.

Construct	Description
ASI	Attends to Specific Ideas
AQTA	Asks Questions Tied to Actions
SWE	Show Work and Explain

Table 1: Summary of elicitation constructs.

Two raters with content analysis expertise participated in the annotation. Each teacher turn was annotated for each of the three constructs. The development set was used for developing rubrics, rater training, joint coding, individual coding, and double coding to monitor agreement. For the test set, one transcript was coded jointly and discussed, while the remaining 19 transcripts (266 teacher turns in all) were coded independently to evaluate inter-annotator agreement. The coding was reliable, achieving Cohen's κ of 0.88, 0.86, and 0.93, for AQTA, ASI, and SWE, respectively.

Table 2 exemplifies the flow of the conversation and the annotations of the teacher's utterances.

The first four teacher utterances contain at least one effective elicitation move each. The first two include asking Cecilia about something that she has done (provided the answer 8 in utterance#1, drew circles in utterance#3; AQTA = 1), specifically to explain the process ('how did you...') or reasoning ('why did you...'), respectively (SWE = 1). The third teacher's utterance (#5 in the Table) includes asking for Cecilia's reasoning about what she has just done (AQTA = 1, SWE = 1), following up

U#	Speaker	Utterance	Classifications		
			ASI	AQTA	SWE
1	Teacher	Cecilio, how did you come up with the answer of eight?	0	1	1
2	Cecilia	When we asked. About the crayons. I drew circles for the crayons. Mike had 6 and Ann had 2. I counted 6,7,8.			
3	Teacher	That's great. I noticed. I noticed when you drew your circles that you have three circles. With a dot inside of them. Why did you put the dot inside those three circles?	0	1	1
4	Cecilia	I put dots to show where I started counting Ann's crayons.	1	1	1
5	Teacher	So why do you have. Three dots instead of just two dots. If Anne only has two crayons.			
6	Cecilia	The first dot is where I started counting from Mike's crayons.	1	0	0
7	Teacher	Do you worry that you might get confused by having three dots to represent Anne's two? Crayons.			
8	Cecilia	Nope! I know which dots go to Ann's crayons because I still count them starting at Mike's crayons.	0	0	0
9	Teacher	You were able to come up. With the correct answer, Cecilia. But I worry in the future. Your extra dot might cause some confusion.			
10	Cecilia	Okay. I'll try to remember that.	0	0	0
11	Teacher	Maybe. A helpful strategy going forward would be to put the number one. In the circle where you started instead of using dots.			
12	Cecilia	Okay, I'll try that!			

Table 2: An excerpt from a conversation between the Teacher (human participant) and Cecilia (the GenAI powered student). Data is shown as it appears in the automated transcription. Human classifications of each teacher utterance for the three constructs are shown in the last three columns.

on Cecilia's idea of putting the dots where she starts counting Ann's crayons (ASI = 1). The next teacher utterance continues the follow-up on Cecilia's three dots idea (ASI = 1), though not about a specific action just performed (AQTA = 0) and not asking a process or reasoning question, but an attitude question ('do you worry...', SWE = 0). The last two teacher utterances do not contain any of the effective moves. The final utterance, specifically, is the teacher's attempt to teach Cecilia to improve her strategy for adding – which is out of scope for the simulation that focuses on eliciting the student's ideas rather than on changing them. A more effective elicitation strategy would have been to move on after the first three exchanges and ask Cecilia to solve a different problem that would help the teacher further explore Cecilia's mathematical understanding, for example, asking her to add more than two numbers or bigger numbers, or checking if she is familiar with the commutative property by solving 2+6 instead of 6+2.

4.3 Datasets

Table 3 shows the descriptive statistics of the data.

Development set				Test set		
#Pa	40			10		
#Tr	40			20		
#Ut	630			275		
#W	11K			3.6K		
	ASI	AQTA	SWE	ASI	AQTA	SWE
%1	28	26	23	27	40	29

Table 3: Descriptive statistics of the datasets. #Pa: Participants. #Tr: Transcripts. #Ut: Teacher utterances. #W: Total word count of all teacher utterances. %1: Percentage of teacher utterances labeled as class 1.

5 Models for evaluating teacher discourse

The goal of the evaluation models is to power formative feedback to teachers immediately following the simulation. Two types of feedback are created. First, for every construct, we show up to 3

utterances where the teacher engaged in the target behavior and up to 3 utterances where they did not. As an additional resource, examples from other teacher’s performances are provided. Second, for every construct, we show the overall frequency of occurrence of utterances where the target behavior occurred, and compare it to the range calculated using data from well-performing teachers. To support both types of feedback, the evaluation models are implemented as classifiers. For each construct, we classify each teacher utterance as showing the target behavior (class 1) or not (class 0).

5.1 Baselines

5.1.1 Few-shot models

Each construct was evaluated through LLM prompting using Azure OpenAI GPT-4o structured outputs. Each prompt instructed the model to identify and extract exact conversational turns from the transcript containing evidence for a single construct. Structured output returned verbatim excerpts with original role (i.e., student or teacher), message string, and turn identifier fields, enabling turn-level binary classification downstream. An empty list was returned when no evidence was found. All prompts were constructed as single system messages containing task instructions, the standardized problem context, construct-specific requirements with between two and five positive and negative examples picked by the subject matter expert on the team, and the full chat history transcript. Appendix B provides the prompts.

5.1.2 Fine-tuned baseline

As a baseline encoder model, we used DeBERTa (He et al., 2021) model deberta-v3-base¹ (He et al., 2023) from HuggingFace’s Transformers library (Wolf et al., 2020). We picked DeBERTa as it was the latest widely used iteration of BERT-style family of models (at the time of analysis) that achieved state-of-the-art performance on natural language understanding benchmarks like MNLI (Williams et al., 2018) and SQuAD v2.0 (Rajpurkar et al., 2018), improving on both BERT and RoBERTa models (He et al., 2021). The model has 12 layers and a hidden size of 768, 86M backbone parameters with a vocabulary of 128K tokens. This model was pre-trained using 160GB of data. For fine-tuning for our classification task, the learning rate was set to 5e-5, trained for 20 epochs. Only the

current teacher utterance were used as input to classify the utterance. We fine-tuned the model on data from four development set folds and evaluated on the fifth fold, in a 5-fold cross-validation design.

5.2 Experimental model

Using the development set, we explored a number of strategies for improving the baseline fine-tuned classifier. We used 5-fold cross-validation on the development set to evaluate the effectiveness of the strategies, separately and in combination. We discuss each strategy in turn.

Weighted loss to account for category imbalance (WB): As can be seen in Table 3, the class distribution is somewhat skewed, with class 1 accounting for fewer than 30% of the teacher utterances for each construct in the development data. In weighted loss, the weights were inversely proportional to the class frequencies.

Context (CO): Prior research has shown that incorporating a limited context of a few utterances can be effective (Tran et al., 2024; Cao et al., 2025). We experimented with a sliding window of three utterances, from three utterances proceeding the current one, two proceeding and one subsequent, all the way to three subsequent utterances as context. Based on the results on the development set, we selected the scheme with one utterance proceeding the current one (the student’s utterance right before the current teacher utterance) and two utterances following the current one (the student’s response and the teacher’s next utterance) as context.

Pre-training using real classroom discussion transcripts (CD): In order to attune the model to teacher-student educational discourse, we pre-trained on the publicly available data of classroom discussion transcripts from the NCTE corpus (Demszky and Hill, 2023). The corpus contains 1,660 transcripts of 45-60 minute 4th and 5th grade mathematics lessons collected by the National Center for Teacher Effectiveness. Cao et al. (2025) found that pre-training on classroom discussion data was useful for developing better classifiers of the same talk moves in tutorial dialogues. Our data isn’t tutoring, in the sense that the teacher’s goal isn’t to help the student learn, but to elicit the students’ current state of knowledge; nevertheless, classroom discourse in the domain of mathematics could be useful pre-training material. We experimented with 1, 5, 10, and 15-epoch pre-training and found that 10-epoch pre-training was the most effective.

¹<https://huggingface.co/microsoft/deberta-v3-base>

Multi-label classification (ML): We observed that while the three teaching constructs capture different aspects of a high quality teacher performance in the simulation, they sometimes happen in the same teacher utterance; see, for example, the conversation excerpt shown in Table 2. Quantitatively, we found that Pearson’s correlation between the list of binary labels for AQTA and SWE across the 630 teacher utterances in the 40 development set simulations was $r = 0.59$, suggesting a tendency to occur in the same utterances. The other correlations were weaker but still significant: $r = 0.37$ for AQTA and ASI and $r = 0.25$ for ASI and SWE. We implemented a multi-label classification with zero to three labels for each teacher utterance.

Selecting the final model for benchmarking: Table 4 shows the results for each strategy on its own (rows WB through CD), using 5-fold cross-validation on the development set. In addition to the four evaluation metrics (see section 5.3) for each of the three constructs, yielding $4 \times 3 = 12$ numbers for each model, we also provide a summary evaluation in the form of the count of cases, out of 12, where the result was substantially better ($\uparrow$) or worse ($\downarrow$) than the fine-tuned baseline, using a change of 0.04 or more as a cutoff. We observe that all strategies were net-positive, namely, increases outnumbered decreases, but (a) all incurred some decreases, and (b) the best net result was only 2 out of 12, suggesting overall mild positive impact. We experimented with combining strategies and picked the best combination, shown in row F(final): The model that included pre-training on the classroom (NCTE) data, multi-label learning, and context. That model achieved a net effect of 8 out of 12, without incurring any decreases. We call this model **FT-Exp** henceforth and benchmark it against the baselines on the test set to evaluate whether the improvement potential is realized on a new set of transcripts unseen during development.

5.3 Model evaluation

To evaluate the models for their ability to identify teacher utterances that realize the given teaching construct, we use both a retrieval paradigm (precision, recall, and F1 scores for the rarer positive class) and a classification paradigm (Cohen’s κ).

For each of the fine-tuned models (FT-BL and FT-Exp), we used five models that we fine-tuned on four out of the five cross-validation folds each. We compute the evaluation metrics for each model on the test set and show the average performance

Mod.	Met.	ASI	AQTA	SWE	$\uparrow$	$\downarrow$
BL	Prec.	.54	.74	.61		
	Rec.	.58	.72	.72		
	F1	.55	.73	.64		
	κ	.37	.62	.52		
WB	Prec.	.58	.68	.58	1	1
	Rec.	.61	.79	.75	1	
	F1	.56	.73	.65		
	κ	.41	.61	.52	1	
CO	Prec.	.58	.66	.63	1	1
	Rec.	.60	.64	.72		1
	F1	.59	.72	.65	1	
	κ	.42	.60	.54	1	
ML	Prec.	.52	.70	.66	1	1
	Rec.	.70	.76	.70	2	
	F1	.58	.71	.65		
	κ	.39	.60	.54		
CD	Prec.	.55	.69	.64		1
	Rec.	.73	.83	.67	2	1
	F1	.62	.74	.62	1	
	κ	.44	.63	.50	1	
F	Prec.	.58	.76	.67	2	
	Rec.	.70	.76	.69	2	
	F1	.61	.76	.67	1	
	κ	.44	.66	.56	3	

Table 4: Dev-set performance of the baseline (BL), each strategy on its own (WB: Weight Balance; CO: Context; ML: Multi-Label; CD: Classroom Discourse) and the final selected model (F). The last two columns count the number of metrics with increase ($\uparrow$) and decrease ($\downarrow$) of at least 0.04 wrt BL.

as the final test set evaluation. In addition, we show the evaluation metrics on the test set for the best aggregate model, as explained in the next section.

5.4 Robustness through aggregation

Due to non-determinism, it is common practice to train multiple times and aggregate the results. It is also possible to intentionally diversify the models by training multiple models on various sub-samples of the data. Aggregation of predictions from multiple such models could improve robustness. Requiring more models to agree is likely to improve precision but hurt recall, while using a permissive aggregation, e.g., positive if at least one model predicts positive, could help recall but hurt precision.

To gauge the optimal aggregation strategy, one ideally would train multiple models based on different subsets of the fine-tuning data for each of the cross-validation folds, resulting in $5 \times 5 = 25$ dif-

ferent models. This would require sub-partitioning of each fold into 80% and 20% again; given that a fold already contains only 8 transcripts, this design would further reduce the training data and result in a small evaluation set for the aggregation models. We therefore opted for a different design to gauge the optimal aggregation strategy: We randomly picked one of the FT-Exp models fine-tuned on four development folds and re-tuned it with four different random seeds. This created some variation in the models, and we considered how many positive votes we should require for the best balance of precision and recall. The results are shown in Table 6 in Appendix A. The results showed that while the most permissive model has the best recall and the most conservative had the best precision, as expected, the best precision and recall combination is achieved by a majority vote – requiring three out of the five models to assign the teacher utterance to class 1 in order for the aggregate to be class 1.

Based on these findings, for the test set, we will provide the results of the majority aggregation of five models (Agg FT-Exp), each fine-tuned on four out of five development set folds.

6 Results

Table 5 shows the results. We observe that fine-tuning was quite effective, the small number of transcripts notwithstanding. Both FT models improved over the Few Shot baseline on all metrics, on average across the three target constructs.

Next, we observe that FT-Exp gained in recall (+0.12) more than it lost in precision (-0.06) with respect to FT-BL, on average across constructs, resulting in improvements in F1 and κ scores, both on average and for each of the constructs.

Let us next compare the results for FT-Exp, shown as average performance on the test set across five models trained on four out of five development folds, to Agg FT-Exp, namely, a single system that aggregates the classifications generated by the five systems above using a majority vote. The results show clearly that the aggregated model is similar or better than the average single model.

7 Discussion

Overall, there is no reason to prefer few-shot (FS) models over fine-tuned (FT) ones, as the only cases of lower scores for FT models are recall for ASI for FT-BL and precision for ASI for FT-Exp, but, in both cases, these were compensated by a larger in-

Model	Metric	ASI	AQTA	SWE	Av.
Few Shot	Prec.	.53	.65	.73	.64
	Recall	.45	.59	.61	.55
	F1	.49	.62	.67	.59
	κ	.29	.41	.53	.41
FT BL	Prec.	.68	.82	.71	.74
	Recall	.41	.71	.70	.61
	F1	.48	.75	.71	.65
	κ	.36	.60	.59	.52
FT Exp.	Prec.	.50	.85	.70	.68
	Recall	.71	.72	.77	.73
	F1	.58	.78	.73	.70
	κ	.38	.64	.61	.54
Agg. FT Exp.	Prec.	.54	.84	.70	.69
	Recall	.81	.77	.78	.79
	F1	.65	.81	.74	.73
	κ	.48	.69	.60	.59

Table 5: Test-set performance of the two baselines and the best experimental models.

crease in precision and recall, respectively, showing overall gains in κ and similar or better F1 scores. For AQTA and SWE, the FT-Exp model crossed the $\kappa \geq 0.60$ threshold that is often taken to be sufficient for formative purposes, showing that fine-tuning, even with relatively little data, can result in usable models. That said, one of the constructs – ASI – remained challenging for all the models.

The FT-Exp model is the most promising, not only due to quality of performance, but also in terms of maintenance and cost of exploitation. Since FT-Exp employs multi-label learning, there is just one model dealing with all the three constructs. Being a relatively small BERT-family model, it can be installed locally and avoid incurring cost per query, as do the larger and more powerful LLMs, like GPT-4o used here for the few-shot model.

Our results also suggest that it may be beneficial to introduce some variation in the models by fine-tuning multiple models on somewhat different subsets of the fine-tuning data. In particular, majority aggregation of five such models results in improved recall across the board, without paying a price in precision. A different aggregation strategy to prioritize recall or precision could be devised, as well as optimizing the number of models and the extent of difference between them; more development data would be necessary for this optimization task. Our results show that a strategy evaluated on development set using a model variation method

that does not require data sub-partitioning resulted in a robust aggregation strategy that generalized to a blind test set with a different model variation method. While using multiple models to attain more robust performance is not a new finding in the LLM world, our results show that model aggregation can be successfully developed and evaluated using relatively little data.

Limitations

Although our results cover multiple effective elicitation moves, they are all studied in the context of a single task. Additional elicitation tasks in upper elementary mathematics and science are currently under development. We intend to replicate the study design when early data become available.

In addition, while the test set contained data from the target population of users and was collected in the target context of use of the simulation, the data come from a small number of pre-service teachers. As the simulation is used in subsequent data collections, we will examine the performance of the automated evaluation models on additional data to gauge the model’s generalization.

We note that the focus of the current paper is the evaluation of teacher performance in the simulation. The usefulness of an evaluation like this is related but not fully determined by the simulation itself constituting an effective practice based *learning* opportunity for educators. In particular, even if the simulation has some deficiency in terms of the realism of the simulated student’s discourse (Scarlatos et al., 2026), automated evaluation of the teacher’s performance could still be an efficient way to gauge the teacher’s current skill with elicitation strategies, if validated for this purpose. We thank a BEA reviewer for bringing this issue up.

Due to team capacity limitation, we explored only one of the most powerful LLMs available (at the time when the few-shot baselines were developed, GPT 4.0 was the top model). It is possible that few shot learning results would be better with the newer GPT model and/or with other LLMs such as Claude. It is also possible that open-source LLMs such as LLama can also be effectively fine-tuned in a parameter-efficient way (Wang et al., 2025), though, as Tran et al. (2024), we note that the data sizes are most likely too small to move the needle for these models. We leave it to future work to explore other LLMs, especially for the ASI construct for which we do not yet have a satisfac-

tory model. We believe our findings regarding the effectiveness of pre-training using classroom data, of flexible context that may include looking ahead in the conversation, of utilizing the teacher’s ability to include more than one effective strategy in a single utterance, and model variation and aggregation methods are a good starting point for further research in automated evaluation of teacher performance.

Ethics statement

The data collections that yielded the data used in this study were conducted with the approval of the IRB of the authors’ institution.

Feedback to the user in newly developed tasks is an inherently complicated proposition. This is because for the feedback to be most pedagogically appropriate, it needs to take into account how users typically behave in the activity, what strong and weak performances look like, and what properties of the performances best differentiate between the two. This is very hard to do with only usability data that may not even come from the target demographic and/or the target use settings. Therefore, provision of feedback is often delayed until well-performing feedback models can be built, which may deprive the early users of some of the potential benefit of participating in the simulation. Our results are quite optimistic in this respect: With only 40 development transcripts obtained through a usability study, we attained acceptable performance for two out of the three target constructs, suggesting that feedback along these two dimensions can now be responsibly provided to subsequent users.

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 Innovative Use of NLP for Building Educational Applications (BEA)*, pages 224–233.
- Yuya Asano, Beata Beigman Klebanov, and Jamie Mikeska. 2025. [Exploring task formulation strategies to evaluate the coherence of classroom discussions with GPT-4o](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 716–736, Vienna, Austria. Association for Computational Linguistics.
- Beata Beigman Klebanov, Michael Suhan, and Jamie Mikeska. 2025. [Towards evaluating teacher discourse without task-specific fine-tuning data](#). In

- Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Full Papers*, pages 192–200, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- 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.
- Jie Cao, Abhijit Suresh, Jennifer Jacobs, Charis Clevenger, Amanda Howard, Chelsea Brown, Brent Milne, Tom Fischhaber, Tamara Sumner, and James H. Martin. 2025. [Enhancing talk moves analysis in mathematics tutoring through classroom teaching discourse](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7671–7684, Abu Dhabi, UAE. Association for Computational Linguistics.
- 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 Thomas, Susan Stansberry, and Ying Xiu. 2020. A mixed reality simulation offers strategic practice for pre-service teachers. *Computers & Education*, 144:103696.
- 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, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. Measuring conversational uptake: A case study on student-teacher interactions. *arXiv preprint arXiv:2106.03873*.
- 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 Dieker, Carrie Straub, Michael Hynes, Charles Hughes, Caitlyn Bukathy, Taylor Bousfield, and Samantha Mrstik. 2019. Using virtual rehearsal in a simulator to impact the performance of science teachers. *International Journal of Gaming and Computer-Mediated Simulations*, 11(4):1–20.
- Patrick J Donnelly, Nathaniel Blanchard, Andrew M Olney, Sean Kelly, Martin Nystrand, and Sidney K D’Mello. 2017. Words matter: Automatic detection of teacher questions in live classroom discourse using linguistics, acoustics, and context. In *Proceedings of the 7th International Learning Analytics & Knowledge Conference*, pages 218–227.
- Zara Ersozlu, Susan Ledger, and Linda Hobbs. 2021. Virtual simulation in ITE: technology driven authentic assessment and moderation of practice.
- Anne Estapa and Jeni Davis. 2023. Prospective teachers’ instructional decisions and pedagogical moves when responding to student thinking in elementary mathematics and science lessons. *International Journal of Science and Mathematics Education*, 21(5):1703–1724.
- Ingólfur Gíslason. 2026. Learning about human mathematical dialogue from dialogue with chatbots: Babbaling, gargling and funnelling. *The Mathematics Enthusiast*, 23(1):167–184.
- Allyson Hallman-Thrasher, Ginger Rhodes, and Kyle Schultz. 2020. Supporting mathematics teacher educators’ practices for facilitating prospective teachers’ mathematical explanations in content courses. *The Mathematics Enthusiast*, 17(2):883–906.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTav3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *International Conference on Learning Representations*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced BERT with disentangled attention](#). In *International Conference on Learning Representations*.
- Sumin Hong, Jewoong Moon, Taeyeon Eom, Idowu David Awoyemi, and Juno Hwang. 2025. Generative AI-enhanced virtual reality simulation for pre-service teacher education: A mixed-methods analysis of usability and instructional utility for course integration. *Education Sciences*, 15(8):997.
- Annemarie Horn, Marcia Rock, Laura Chezan, Jonna Bobzien, Olga Karadimou, and Aeshah Alturki. 2023. Effects of e coaching on the occurrence, equity, and variety of behavior specific praise during Mursion™ simulations. *Journal of Special Education Technology*, 38(4):501–514.
- Ruikun Hou, Tim Fütterer, Babette Bühler, Efe Bozkir, Peter Gerjets, Ulrich Trautwein, and Enkelejda Kasneci. 2024. Automated assessment of encouragement and warmth in classrooms leveraging multimodal emotional features and ChatGPT. In *International Conference on Artificial Intelligence in Education*, pages 60–74. Springer.
- Nicholas Hunkins, Sean Kelly, and Sidney D’Mello. 2022. “beautiful work, you’re rock stars!”: Teacher analytics to uncover discourse that supports or undermines student motivation, identity, and belonging in classrooms. In *Proceedings of the 12th International Learning Analytics & Knowledge Conference*, pages 230–238.

- Michael Ilagan, Beata Beigman Klebanov, and Jamie Mikeska. 2024. [Automated evaluation of teacher encouragement of student-to-student interactions in a simulated classroom discussion](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 182–198, Mexico City, Mexico. Association for Computational Linguistics.
- 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, page 1–13, New York, NY, USA. Association for Computing Machinery.
- Sean Kelly, Andrew M. Olney, Patrick Donnelly, Martin Nystrand, and Sidney K. D’Mello. 2018. [Automatically measuring question authenticity in real-world classrooms](#). *Educational Researcher*, 47(7):451–464.
- Ashlee Kupor, Candice Morgan, and Dorottya Demszky. 2023. Measuring five accountable talk moves to improve instruction at scale. *arXiv preprint arXiv:2311.10749*.
- Dabae Lee, Taekwon Son, and Sheunghyun Yeo. 2025. Impacts of interacting with an AI chatbot on preservice teachers’ responsive teaching skills in math education. *Journal of Computer Assisted Learning*, 41(1):e13091.
- Dabae Lee, Corey Webel, Sheunghyun Yeo, Burcu Alapala, and Rebekah Hanak. 2026. Cultivating responsive teaching with AI: exploring preservice teachers’ questioning patterns with student-emulating agents. *Journal of Mathematics Teacher Education*.
- Tammy Lee, Carrie Lee, Mark Newton, Paul Vos, Jennifer Gallagher, Daniel Dickerson, and Camryn Regenthal. 2024. Peer to peer vs. virtual rehearsal simulation rehearsal contexts: Elementary teacher candidates’ scientific discourse skills explored. *Journal of Science Teacher Education*, 35(1):63–84.
- Jieun Lim, Unggi Lee, Junbo Koh, Yeil Jeong, Yunseo Lee, Gyuri Byun, Haewon Jung, Yoonsun Jang, Sanghyeok Lee, and Jewoong Moon. 2025. Development and implementation of a generative artificial intelligence-enhanced simulation to enhance problem-solving skills for pre-service teachers. *Computers & Education*, 232:105306.
- Bojie Liu, Yu Yang, and Guandong Xu. 2026. GenAI-empowered virtual micro-teaching training system for preservice teachers. In *Behavioural and Social Computing*, pages 232–241, Singapore. Springer Nature Singapore.
- 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](#). *Preprint*, arXiv:1907.11692.
- Pamela Lottero-Perdue, Karen Cimino, and Julia Brandeberry. 2023. Preservice teachers facilitating a discussion with elementary student avatars before facilitating it with real students. *Innovations in Science Teacher Education*, 8(4).
- Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. [MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.
- Julia Markel, Steven Opferman, James Landay, and Chris Piech. 2023. GPTEach: Interactive TA training with GPT-based students. In *Proceedings of the 10th ACM conference on Learning@ Scale*, pages 226–236.
- Luis Marquez-Carpintero, Alberto Lopez-Sellers, and Miguel Cazorla. 2025. Simulating students with large language models: A review of architecture, mechanisms, and role modelling in education with generative AI. *arXiv preprint arXiv:2511.06078*.
- Sarah Michaels and Catherine O’Connor. 2015. Conceptualizing talk moves as tools: Professional development approaches for academically productive discussion. *Socializing intelligence through talk and dialogue*, pages 347–362.
- Jamie Mikeska, Dionne Cross Francis, Pamela Lottero-Perdue, Meredith Park Rogers, Calli Shekell, Pavneet Kaur Bharaj, Heather Howell, Adam Maltese, Meredith Thompson, and Justin Reich. 2025a. Promoting preservice teachers’ facilitation of argumentation in mathematics and science through digital simulations. *Teaching and Teacher Education*, 154:104858.
- Jamie Mikeska, Heather Howell, and Devon Kinsey. 2023a. Do simulated teaching experiences impact elementary preservice teachers’ ability to facilitate argumentation-focused discussions in mathematics and science? *Journal of Teacher Education*, 74(5):422–436.
- Jamie Mikeska, Beata Beigman Klebanov, Aakanksha Bhatia, Shreyashi Halder, and Michael Suhan. 2025b. Evaluating the use of generative artificial intelligence to support learning opportunities for teachers to practice engaging in key instructional skills. In *Artificial Intelligence in Education*, pages 378–391, Cham. Springer Nature Switzerland.
- Jamie Mikeska, Jonathan Steinberg, Pamela Lottero-Perdue, Dante Cisterna, Devon Kinsey, and Heather Howell. 2023b. Using simulated classrooms to examine elementary teachers’ perceptions about, attention to, and use of formative feedback to improve their ability to facilitate science discussions. *Contemporary Issues in Technology and Teacher Education*, 23(1):48–83.

- Tanya Nazaretsky, Jamie Mikeska, and Beata Beigman Klebanov. 2023. Empowering teacher learning with AI: Automated evaluation of teacher attention to student ideas during argumentation-focused discussion. In *Proceedings of the 13th International Learning Analytics and Knowledge Conference*, pages 122–132.
- Sitong Pan, Robin Schmucker, Bernardo Garcia Bulle Bueno, Salome Aguilar Llanes, Fernanda Albo Alarcón, Hangxiao Zhu, Adam Teo, and Meng Xia. 2025. TutorUp: What if your students were simulated? training tutors to address engagement challenges in online learning. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18.
- John Pecore, Corey Nagle, Tadlee Welty, Minkyoun Kim, and Melissa Demetrikopoulos. 2023. Science teacher candidates’ questioning and discussion skill performance in a virtual simulation using experiential deliberate practice. *Journal of Science Teacher Education*, 34(4):415–435.
- Yi Qi and Gary Sykes. 2016. [Eliciting student thinking: Definition, research support, and measurement of the ETS® National Observational Teaching Examination \(NOTE\) Assessment Series](#). *ETS Research Memorandum No. RM-16-06*.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for squad](#). *Preprint*, arXiv:1806.03822.
- Borhan Samei, Haiying Li, Fazel Keshtkar, Vasile Rus, and Arthur C Graesser. 2014. Context-based speech act classification in intelligent tutoring systems. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, pages 236–241. Springer.
- Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. 2026. Simulated students in tutoring dialogues: Substance or illusion? *arXiv preprint arXiv:2601.04025*.
- Danner Schlotterbeck, Pablo Uribe, Abelino Jiménez, Roberto Araya, Johan van der Molen Moris, and Daniela Caballero. 2021. Tarta: Teacher activity recognizer from transcriptions and audio. In *Artificial Intelligence in Education*, pages 369–380, Cham. Springer International Publishing.
- Meghan Shaughnessy and Timothy Boerst. 2018. Uncovering the skills that preservice teachers bring to teacher education: The practice of eliciting a student’s thinking. *Journal of Teacher Education*, 69(1):40–55.
- Taekwon Son. 2025. Noticing classes of preservice teachers: Relations to teaching moves through AI chatbot simulation. *Education and Information Technologies*, 30(7):9161–9184.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James Martin, and Tamara Sumner. 2022. The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 4654–4662.
- Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2024. [Multi-dimensional performance analysis of large language models for classroom discussion assessment](#). *Journal of Educational Data Mining*, 16(2):304–335.
- Deliang Wang, Yaqian Zheng, Jinjiang Li, and Gaowei Chen. 2025. Parameter-efficiently fine-tuning large language models for classroom dialogue analysis. *IEEE Transactions on Learning Technologies*, 18:542–555.
- Rose Wang and Dorottya Demszky. 2023. Is ChatGPT a Good Teacher Coach? Measuring Zero-Shot Performance For Scoring and Providing Actionable Insights on Classroom Instruction. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 626–667.
- Liwei Wei, Karen Murphy, and Carla Firetto. 2018. How can teachers facilitate productive small-group talk? An integrated taxonomy of teacher discourse moves. *The Elementary School Journal*, 118(4):578–609.
- Jacob Whitehill and Jennifer LoCasale-Crouch. 2024. Automated evaluation of classroom instructional support with LLMs and BoWs: Connecting global predictions to specific feedback. *Journal of Educational Data Mining*.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). *Preprint*, arXiv:1704.05426.
- 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, and 3 others. 2020. [Hugging-face’s transformers: State-of-the-art natural language processing](#). *Preprint*, arXiv:1910.03771.
- Dawn Woods, Linda Doornbos, and Cynthia Carver. 2024. Using mixed reality simulations to prepare justice-oriented teachers. *Contemporary Issues in Technology and Teacher Education*, 24(1):95–124.
- Paiheng Xu, Jing Liu, Nathan Jones, Julie Cohen, and Wei Ai. 2024. The promises and pitfalls of using language models to measure instruction quality in education. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4375–4389.

Junyan Zhang, Yiming Huang, Shuliang Liu, Yubo Gao, and Xuming Hu. 2025. Do BERT-like bidirectional models still perform better on text classification in the era of LLMs? In *Findings of the Association for Computational Linguistics: EMNLP*, page 18980–18989. Association for Computational Linguistics.

Nuodi Zhang, Fengfeng Ke, Chih-Pu Dai, Sherry Southerland, and Alex Barrett. 2026. [Science and mathematics preservice teachers’ perceptions and experiences of practicing dialogic teaching in generative AI-powered virtual reality simulation](#). *Teaching and Teacher Education*, 171:105349.

Yuling Zhuang and Si Zhang. 2025. Pre-service mathematics teachers’ perceptions of using GenAI for practicing teacher questioning: A semester-long study. *Eurasia Journal of Mathematics, Science and Technology Education*, 21(9):em2689.

Liu Ziyang, Hu Yongchun, and Zhang Qiaoping. 2026. [Harnessing artificial intelligence for preservice teachers’ development: A scoping review of applications, benefits, and challenges](#). *Computers and Education Open*, 10:100330.

A Aggregation Schemes Evaluation

#1s	Metric	ASI	AQTA	SWE	Av.
1	Prec.	.53	.68	.65	.62
	Recall	.78	.81	.82	.80
	F1	.63	.74	.72	.70
2	Prec.	.55	.70	.66	.64
	Recall	.77	.80	.80	.79
	F1	.64	.75	.73	.71
3	Prec.	.57	.71	.67	.65
	Recall	.75	.80	.80	.78
	F1	.65	.75	.73	.71
4	Prec.	.58	.71	.67	.65
	Recall	.72	.80	.78	.77
	F1	.64	.75	.72	.70
5	Prec.	.57	.73	.67	.66
	Recall	.65	.78	.75	.73
	F1	.60	.76	.71	.69

Table 6: Development set performance of aggregation schemes for five FT-Exp models fine-tuned with different seeds. The first column shows the number of class 1 classifications across the five models for the aggregate to be class 1. Thus, models requiring only one class 1 vote have the highest recall (0.80, on average across the three constructs), whereas models requiring full agreement - five class 1 votes - have the highest precision (0.66, on average across the three constructs).

B Feedback Evaluation Module

The feedback module evaluates educator–student conversation transcripts against pedagogical criteria using structured output parsing. A single LLM call is issued per elicitation construct (see Table 1), with the model parameters shown in Table 7.

Table 7: Model parameters for the feedback module

Parameter	Value
Temperature	0
Max Tokens	16,000
Seed	42
Response Format	Structured (Evidence)
Model	GPT-4o (v2024-08-06)

Note. The Evidence response format is a structured output schema containing a list of evidence sequences, each comprising consecutive conversational turns with role, content, and turn identifier fields. The same model deployment and parameters are used for all constructs.

For each construct, the prompt is sent as a single system message. The prompt follows a shared template: a task description specifying the construct name, a shared math problem context describing the student’s work, construct-specific evidence requirements with examples and non-examples, and a transcript placeholder where the full conversation history is inserted at runtime. The response is parsed into a structured Evidence object via the OpenAI structured output API.

The task description is as follows:

Your task is to collect evidence on whether the educator’s questions, prompts, and tasks are conducive to eliciting student thinking in the following criterion: *[construct name]* Return your answer as an Evidence object with lists of examples from the user (educator) from the transcript. Provide excerpts from chat history to give sufficient context for evidence. Follow the examples closely, they must be nearly identical. Return an empty list if nothing is similar to the given examples. All evidence must be written exactly as is from the transcript. Never invent or copy examples.

The shared math problem context included in each prompt is:

The discussion is about how the student solved a problem by completing the following steps on a math worksheet:

1. The student drew 6 circles representing Mike's crayons while saying out loud what they were drawing. The circles are drawn in the same pattern as the six-face of a six-sided dice.
2. The student drew 2 crayons representing Ann's crayons
3. The student continued counting, starting where they left off with Mike's crayons, and continuing by counting Ann's crayons. They did not start counting from 1. They drew a dot on Mike's 6th crayon, and then dots on Ann's two crayons as they counted.
4. After solving the problem, the student wrote the numbers 6, 2, and 8 in boxes on the worksheet to complete $6 + 2 = 8$.

The construct-specific sections of each prompt are presented in sections [B.1](#)- [B.3](#). For evaluation purposes, all and only teacher (User) utterances that are returned as part of the evidence for a particular construct are labeled 1 for that construct.

B.1 Attends to Specific Ideas

Evidence Requirements

- To count as evidence for this criterion, the teacher needs to ask a follow up probing question based on a specific idea that the student shared about their problem solving process or math understanding (could be about the given math problem or new problems that the teacher includes in the conversation). The follow up probing question should require the student to explain why they engage in specific steps or to explain their understanding related to the math problem or concepts.
- If it is the first time that the teacher is asking about something the student did (e.g., Why did you draw three circles with dots in them?), that would not count as evidence. But a follow up question based on the student's initial response (e.g., about the three circles with dots)

would count as evidence. Essentially the student needs to say something about a specific idea before the teacher's follow up prompt can be tagged as evidence for this criterion.

- The follow up prompt does not need to be linked to the immediate preceding utterance. The teacher could be following up on something that the student said earlier in the conversation and that would still count as evidence for this criterion.

Examples of Evidence

Example 1:

User: I see that there are eight crayons in all. Can you tell me why you put a dot in three of the grains.

Assistant: I didn't put a dot in three, I put them on the last three I counted.

User: Okay. So how many crayons did Mike have?

Assistant: 6 crayons.

User: So you drew six grands for Mike? And how many crayons did Anne have?

Assistant: Ann had two crayons.

User: Ann had two crayons. So why did you put a dot in the last three, the two circles?

Assistant: I put a dot on Mike's last crayon and Ann's two crayons when I counted them to make 7 and 8.

Example 2:

User: What are the dots in the middle of the circles? What do those represent?

Assistant: The dots are for counting.

User: So why did you only count three of them?

Example 3:

User: Okay, I see that. And why are there dots in three of the circles?

Assistant: I put dots when I count the crayons.

User: Okay, so why are there not dots in the first five?

Example 4:

User: Well, how do you know that's how many crowns they have altogether?

Assistant: I drew 6 for Mike and 2 for Ann. Then I counted, 6, 7, 8.

User: And why did you count? Six, seven, eight.

The following would NOT count as evidence:

- If the teacher restates the student's idea without probing further (only restates to clarify that is what the student said).

- If the teacher asks the students to recall the steps they used to solve the given addition word problem.

Non-Examples

Non-Example 1:

User: So if you remember, here is where normally. I would have manipulatives. For example, maybe some markers. Let's pretend that these markers are crayons, like the circles you drew. So. You first drew six circles. Like this for? Mike's crayons, right?

Assistant: Yes.

Non-Example 2:

User: You drew two circles next because Ann has 2 crayons, right?

Assistant: Yes!

B.2 Asks Questions Tied to Actions

Evidence Requirements

- To count as evidence for this criterion, the teacher needs to ask questions about what the student did or about why the student did specific things to solve the given addition word problem or to solve new addition or subtraction number sentences or word problems.
- These questions do not need to link to the student's preceding comment or utterance.
- If the teacher asks the student a question to confirm or clarify about what they did (something that the student already mentioned earlier in the conversation), that would count as evidence for this criterion.

Examples of Evidence

Example 1:

User: So why did you draw eight circles?

Assistant: I drew 6 circles for Mike's crayons and 2 for Ann's crayons.

Example 2:

User: I'd really like to learn too. Can you show me how you're working this? Problem. What's the first step?

Assistant: First, I draw circles.

Example 3:

User: Can you explain to me? Why? You did, the steps that you did.

Assistant: I drew circles to show the crayons for Mike and Ann. Then I counted 6, 7, 8.

Example 4:

User: Okay, so I see three circles. With a dot on the inside. I'm wondering what made you draw a third circle.

Assistant: I drew two circles for Ann and put dots as I counted.

Example 5:

User: Why did you choose that strategy?

Assistant: That's how I show the crayons.

The following would NOT count as evidence:

- If the teacher's question focuses on what the student understands and is not related to something specific they did to solve the given addition word problem or new addition or subtraction number sentences or word problems. For example, "What does addition mean?", "What does it mean to count on?"
- If the teacher's question asks the student to answer a factual (recall) question related to the given addition word problem.
- If the teacher's questions asks the student what they would do in solving additional problems or what they would do in the future. For example, gives them a new problem and asks them to solve it.

Non-Examples

Non-Example 1:

User: What does the plus sign mean?

Assistant: The plus sign means you add things together.

Non-Example 2:

User: What do you know about how commutative property can be used to solve addition problems?

Assistant: I'm not sure.

Non-Example 3:

User: How is addition similar to multiplication?

Assistant: I just know addition.

Non-Example 4:

User: How many crayons did Mike have?

Assistant: Mike had 6 crayons.

B.3 Shows Work and Explains

Evidence Requirements

- To count as evidence for this criterion, the teacher needs to prompt the student to reference or discuss one or more aspects of their

written work AND explain what they did in that written work or why the written work is the way it is.

Examples of Evidence

Example 1:

User: Which crayons were Mike? How did you show Mike's crayons?

Assistant: I drew 6 circles for Mike's crayons.

Example 2:

User: Excellent. So tell me, how did you count on from six?

Assistant: I started at 6 and then counted 7, 8.

Example 3:

User: So what was the first thing you did to solve the problem?

Assistant: I drew 6 circles for Mike's crayons.

The following would NOT count as evidence:

- If the teacher restates an idea that the student already shared.
- If the teacher asks the students to recall the steps they used to solve the given addition word problem
- If the teacher's question asks the student what they would do in solving additional problems or what they would do in the future (e.g., gives them a new problem and asks them to solve it)

Non-Examples

Non-Example 1:

User: Now I can understand. You started at six, and then you added two more, so that's how you got to. Eight. Is that right?

Assistant: Yes.

Non-Example 2:

User: So if you remember, here is where normally. I would have manipulatives. For example, maybe some markers. Let's pretend that these markers are crayons, like the circles you drew. So. You first drew six circles. Like this for? Mike's crayons, right?

Assistant: Yes.

Multi-component student writing profiles for expert-aligned automated evaluation of English learner essays.

Russell Moore
ALTA Institute
Computer Laboratory
University of Cambridge
rjm49@cam.ac.uk

Andrew Caines
ALTA Institute
Computer Laboratory
University of Cambridge
apc38@cam.ac.uk

Paula Buttery
ALTA Institute
Computer Laboratory
University of Cambridge
pjb48@cam.ac.uk

Abstract

Automated Writing Evaluation (AWE) platforms have become common, but a significant gap remains between automated assessment and expert human feedback. We address this gap by introducing a supervised learning method that uses a multi-component student writing profile (comprising estimated CEFR level, grammatical error rates, and vocabulary distribution) to align AI scoring with expert human judgements. In the context of an online essay-writing platform for second language learners of English, our model achieves a 36% reduction in RMSE for holistic essay scoring and an 84% improvement in similarity to human-expert analysis of grammatical errors compared to automarker scores (26% and 57% improvement from the best-performing comparable earlier work, by Zaidi et al., 2019). Furthermore, we demonstrate that the model can predict a student’s profile for a final-version essay from earlier drafts and that predictions generalise to a subsequent task, offering new possibilities for automated curriculum planning. Finally, we introduce a visualisation tool that provides educators with clear expert-aligned longitudinal views of student development.

1 Introduction

Automated Writing Evaluation (AWE) platforms typically allow learners to submit a sequence of revised essays which incorporate system-generated suggestions for improving spelling, vocabulary and grammar. However, these platforms fail to provide the depth of feedback found in human instruction.

Standard metrics for assessing essay quality (e.g., TTR, MTLD, error-free-clause-ratio, grammatical error density, Flesch-Kincaid grade level) offer coarse-grained statistics, but they sacrifice nuance for summarisation. In this work we move beyond such metrics by representing texts as a richer *writing profile* object. This profile comprises sub-components for holistic proficiency, grammatical

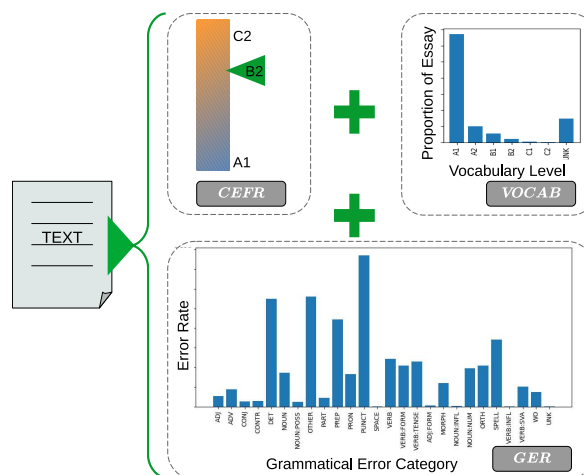


Figure 1: The *writing profile* concept; rather than a single score, a student’s writing is represented by a profile with an estimated CEFR level, grammatical error rates (GER) and a profile of vocabulary usage (VOCAB).

error rates and vocabulary use; this provides a more granular view of student development and captures elements that the student *can do* (vocabulary usage) as well as elements that the student is struggling with (grammatical error rates). The concept is illustrated in Figure 1.

Writing profiles provide a dual-purpose representation that is simultaneously human-interpretable (offering pedagogical reference for educators) and machine-readable (as required for automated scoring). In this work, the profiles integrate a holistic essay score mapped to the CEFR¹ (North, 2005), with granular diagnostic actionable data. Specifically, the grammatical error rates are mapped to 26 error categories that are displayed by the platform, and are actionable by a language teacher; while the vocabulary usage data can be visualised as a histogram depicting vocabulary range and distribution, which aligns with standardised exam-marking

¹<https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions>

rubric requirements.^{2,3} Together the components constitute a multi-dimensional linguistic signature of a learner’s essay, expressing holistic proficiency, specific aptitude and the areas for improvement.

We source our essays from the *Write & Improve Corpus 2024* (WI_CORPUS) (Nicholls et al., 2024), a public dataset sourced from a major online essay practice platform, *Write & Improve*⁴ (Andersen et al., 2013; Yannakoudakis et al., 2018). This dataset comprises sequences of revisions of learner essays. The final version of each essay is scored by a human expert and mapped to a CEFR-level; the essay is also manually corrected for grammatical (morpho-syntactic) errors. The holistic score, the distribution of the errors, and the vocabulary-usage in the corrected version provides a high-fidelity *expert profile* which can be used as the gold-standard for computer-generated *auto profiles*.

Auto profiles derived using popular NLP tools (*Language Tool*, *ERRANT*, *spaCy*) do not align closely with these expert profiles. To calibrate the auto-profiles, and significantly enhance their alignment with the human experts, we train a neural model using the expert profiles as target data. Beyond this calibration task, we demonstrate that the same model can be used to accurately forecast a student’s final submission profile from an initial draft. Furthermore, we show that the model can also generalise, accurately predicting the writing profile for a subsequent task (which opens possibilities for long-term curriculum planning).

In summary, the contributions of this work are as follows: (1) High-fidelity calibration: a model for accurately aligning automated CEFR-level assignments and grammatical error rates with expert judgments; (2) Vocabulary forecasting: predictions of a learner’s exhibition of vocabulary-usage on a task; (3) Predictive modeling: a model to predict a student’s performance in the final version of the current task from an early draft; and (4) Longitudinal predictions: a model to predict a student’s performance in a subsequent task.

²https://takeielts.britishcouncil.org/sites/default/files/ielts_task_2_writing_band_descriptors.pdf

³<https://www.cambridgeenglish.org/images/210434-converting-practice-test-scores-to-cambridge-english-scale-scores.pdf>

⁴<https://writeandimprove.com/>

2 Related Work

Accurate modeling of longitudinal skill acquisition is key for adaptive, personalised tutoring systems (Corbett and Anderson, 1994; Anderson et al., 1995). These systems typically rely on a well-defined taxonomy of skills (or *knowledge components* – Koedinger et al. 2012) with a particular acquisition model such as that in Cen et al. (2006) and Pavlik Jr et al. (2009), although implementations vary widely.

In language learning, knowledge components tend to be less clearly defined than in STEM subjects. Resources such as the *English Vocabulary Profile*⁵ or the *English Grammar Profile* (O’Keeffe and Mark, 2017) describe components in a qualitative way, but not as directly machine-readable rules. Consequently, many AWE platforms prioritise identification of errors over an analysis of correct usage.

The tasks of Grammatical error detection (GED) and correction (GEC) have been tackled using a range of methods including feature-based classification and machine-translation (for a survey see Bryant et al., 2023). There are several widely-used, annotated corpora for this task but the error typing within them is not consistent. We opt to use the *ERRANT* tool that has an independent error typology that can be used with any such annotated corpora (Bryant et al., 2017).

In our work, Automated Essay Scoring (AES) becomes a sub-task of AWE. AES typically refers to the task of predicting a holistic grade or analytic grades for an essay without interpretable diagnostics. Recent preferred methods for AES function as “black boxes” (e.g. Mayfield and Black, 2020 and Mansour et al., 2024). By training a model on writing profiles, which are interpretable (as required by legislation⁶) we remove the black-box limitations but retain the accuracy of neural models.

To our knowledge only Zaidi et al. (2019) has attempted a similar prediction task on comparable data from the *Write & Improve* platform, and our work draws comparisons with that study as part of our evaluation.

⁵<https://englishprofile.org/?menu=english-vocabulary-profile>

⁶<https://www.europarl.europa.eu/topics/en/article/20230601ST093804/eu-ai-act-first-regulation-on-artificial-intelligence>

3 Write & Improve Corpus 2024

The *Write & Improve* platform has a publicly available dataset (WI_CORPUS) used in this work (Nicholls et al., 2024). This logs the essay-writing activities of English language learners, against prompts covering typical topics such as pets, holidays, etc. For each essay submitted, the *Write & Improve* automarker assigns a level on the CEFR scale (see A.3) and gives feedback on errors in the text, inviting the student to correct and resubmit. Hence there is a sequence of submissions for each student-prompt pair. The student chooses when a task is complete and it is time to move on to another prompt.

As well as the platform scores, the WI_CORPUS dataset includes annotations from a human expert for the final version of each essay: this consists of the human-judged CEFR level, and a human-corrected version of the text. This expert information is used to create *expert profiles*, which form the prediction targets for our neural model.

We define our own validation and test folds, removing a subset of samples where the expert human-corrected versions are not currently publicly available. There are 679 users responding to 50 prompts in the filtered dataset, and 23k submissions. 95% of drafting series are between 2 and 16 submissions long, median=3, mean=4.48. The mean Levenshtein ratio (Levenshtein, 1966) for final draft to human-corrected text is 0.96. For all drafts to human-corrected text it is 0.91. Between pairs of drafts it is 0.97. Prompts in the dataset are assigned to one of three difficulty levels: *beginner*, *intermediate*, or *advanced*.

4 Expert Profile Prediction

The main objective (Task 1) for this work is to improve the auto profiles so that they are more in line with the expert profiles. The secondary objective (Task 2) is forward prediction, where we attempt to predict the final draft’s human profile from an earlier version. The tertiary objective (Task 3) is a combination of these – we take the user automarker features from their final attempt at prompt i and try to forward predict the final profile for prompt $i + 1$, to see how well the final user features from prompt i generalise to other prompts on the platform.

4.1 Automatic Writing Profiles

In order to work with writing profiles, we must first extract them. In WI_CORPUS, only the final drafts

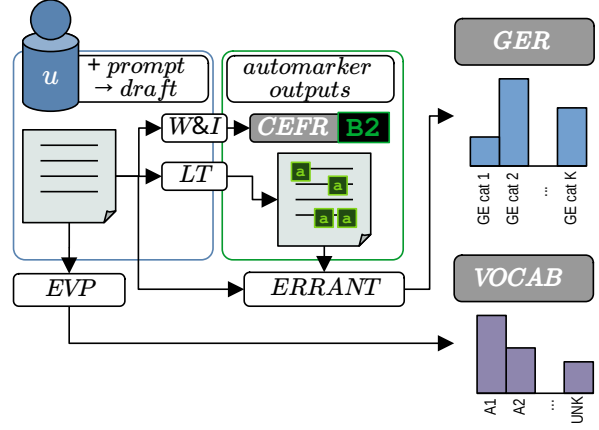


Figure 2: *Auto profile* generation. *W&I* is the *Write & Improve* automarker, *LT* is Language Tool. *EVP* is English Vocabulary Profile. *ERRANT* is the ERRANT grammar annotator. The gray text boxes denote components of the resultant profile. (See section 4.4)

are paired with human CEFR scores and human-corrected texts – these can be used directly in expert profile extraction. For earlier drafts we must approximate some aspects. The dataset already contains an auto-marked CEFR level for every draft. To get an estimated distribution of grammatical errors we must auto-correct the text and compare back to the original using ERRANT (the process is described in section 4.4.3). We also get the distribution of vocabulary used in the draft, by looking up the CEFR level of each content word (see section 4.4.4). These three components together form the *auto profile* of an essay attempt. See Figure 2.

4.2 Neural Network Architecture

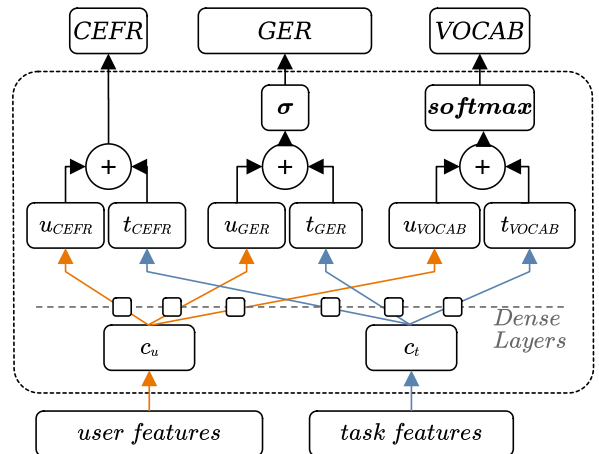


Figure 3: Neural network architecture. A feedforward network with three branches for each component of the predicted output.

The architecture of our neural system can be seen

in Figure 3. The system is feature-based and takes user features $\mathbf{u} = (u_0, \dots, u_n)$ and task features $\mathbf{t} = (t_0, \dots, t_m)$. These features are concatenated for each sample to get vectors $\mathbf{c}_u, \mathbf{c}_t$. For the Grammatical Error Rate predictions, the concatenated inputs are converted into a pair of 26-dimensional vectors by a matrix product, which are then combined as follows:

$$\begin{aligned} u_{GER} &= W_{gu} \cdot D(\mathbf{c}_u) \\ t_{GER} &= W_{gt} \cdot D(\mathbf{c}_t) \\ \widehat{GER} &= \sigma(u_{GER} \oplus t_{GER}) \end{aligned}$$

where $D(\cdot)$ is a dropout layer with possible *rate* $\in [10^{-10}, 1]$ and σ is the *logistic sigmoid* activation function, $\sigma(z) = 1/(1 + e^{-z})$. The elementwise addition $\oplus$ combines the propensity for user error (u_{GER}) with the intrinsic opportunities for error in the exercise (t_{GER}). The sigmoid ensures that these are always in the $[0, 1]$ range – that is, rates are expressed as a per-category *Rasch model* (Rasch, 1980).

For CEFR level, the outputs of the matrix products are a single number, and there is no sigmoid since the target range is $[0, 12]$:

$$\begin{aligned} u_{CEFR} &= W_{cu} \cdot D(\mathbf{c}_u) \\ t_{CEFR} &= W_{ct} \cdot D(\mathbf{c}_t) \\ \widehat{CEFR} &= u_{CEFR} \oplus t_{CEFR} \end{aligned}$$

For the Vocabulary Profile predictions, the histogram estimate is constructed in a similar manner to other outputs, but it is normalised using a *softmax* activation function (Bridle, 1990):

$$\begin{aligned} u_{VOCAB} &= W_{vu} \cdot D(\mathbf{c}_u) \\ t_{VOCAB} &= W_{vt} \cdot D(\mathbf{c}_t) \\ \widehat{VOCAB} &= softmax(u_{VOCAB} \oplus t_{VOCAB}) \end{aligned}$$

The outputs from the neural network are then returned as a triple with element dimensions (26,1,7):

$$OUT = (\widehat{GER}, \widehat{CEFR}, \widehat{VOCAB})$$

Implementation was in *Keras* (Chollet et al., 2015) on *Tensorflow* backend (Abadi et al., 2015) using utilities from *scikit-learn* (Pedregosa et al., 2011).

4.3 Loss functions

System loss is the sum of three loss functions:

4.3.1 Grammar Error Rates

For grammatical error rate vectors we use *Euclidean distance* as a loss to minimise the overall distance between prediction and target:

$$L_{GER} = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_k (x_k - y_k)^2}$$

where the index k spans across the components of the vector, and i spans across the N samples in the dataset. *Mean squared error* loss can be used, but Euclidean distance gives slightly better results.

4.3.2 CEFR Level

CEFR level is modelled as a scalar and for this we use *mean squared error* as a loss:

$$L_{CEFR} = \text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

where x_i is the predicted value, y_i is the actual value, and N is the number of samples.

4.3.3 Vocabulary

Vocabulary use is represented as an empirical probability distribution, so we can use *cross-entropy loss* as an appropriate loss function:

$$L_{VOCAB} = -\frac{1}{N} \sum_{i=1}^N \sum_{c \in C} y_c \log(x_c)$$

where C is the number of classes (*i.e.* CEFR levels), y_c is the true likelihood of class c , and x_c is the predicted likelihood of class c . Both $x_c, y_c \in [0, 1]$ and $\sum_c x_c = \sum_c y_c = 1$.

4.4 Input Features

The WI_CORPUS dataset contains textual and metadata features for student essays. In this section we outline the input features and the pre-processing required to obtain them. For more information on dimensions and types, see A.4.

4.4.1 Auto CEFR Level

This is the CEFR level reported by the *Write & Improve* auto-marker for each essay submission, expressed as an integer in range $[0, 12]$.

4.4.2 Best-essay flags

This *best-essay* feature consists of two binary flags: one indicates the best draft submitted by the user on this task (*task-best*), and one shows the best draft submitted by the user in the whole dataset


```

S Everything seem quite meaningless to me .
C Everything seems quite meaningless to me .
A 0 1 ||R:SPELL||Everything||REQUIRED||-NONE-||0
A 1 2 ||R:VERB:SVA||seems||REQUIRED||-NONE-||0

```

Figure 4: ERRANT-style annotations. S is the original sentence, C is the corrected sentence, Rows marked A are annotations suggested by ERRANT - here a replacement spelling (R:SPELL) and a replacement on the verb to change the subject-verb-agreement (R:VERB:SVA).

(*user-best*). The motivation behind this feature is to provide a signal about the level of this draft relative to previous drafts, on the basis that the task-best and user-best drafts are more reflective of the upper bound of the student’s ability. We use the student’s automarked CEFR history to set these values.

4.4.3 Grammatical Error Rates

A learner’s grammatical ability is expressed as a *grammatical error rates* (GER) object, a vector where each element corresponds to a type of grammatical error. Elements are real-valued and show the per-token rate of occurrence of that error type within a draft. For the main GER objects, the text written by a student is analyzed using *Language Tool* (LT), a widely-used grammatical error detection and correction system (Milkowski, 2010). This system returns a list of probable corrections that are then applied to the student text, to get the LT-corrected text. This is then compared to the original in ERRANT to get the corrections categorised as ERRANT edit codes.

Figure 4 gives an example of ERRANT-style annotation. The red highlights in the first S(entence) line show the words that ERRANT has tagged as edits. The second line is the C(orrection) by a human expert. The third and fourth A(nnotation) lines show token spans for corrections (in blue). The strings highlighted in green are the error types (e.g., R:SPELL is a spelling error; R:VERB:SVA is a subject-verb agreement error).

ERRANTised GER vectors are simplified by collapsing the R/M/U prefixes (for *replace*, *missing*, *unnecessary*) to get 26 categories (see appendix A.2). Note that in the target data, GERs are only available as ERRANT-type vectors, since they are derived from human corrections. ERRANT GER vectors are called **uGER** in the results.

As well as creating ERRANT-style GER vectors, it is possible to create a similar object for LT output directly. Although LT has hundreds of rules to identify errors, it groups them into categories of

which 15 appear in WI_CORPUSS. Counts are summed and normalised as before. LT GER vectors are denoted **uGER.LT** in the results.

4.4.4 Vocabulary Histogram (uVOCAB)

This feature is a histogram of the proportion of words in a draft that fall into each of the CEFR vocabulary difficulty levels, as defined by the *English Vocabulary Profile* (EVP) (Capel, 2015). These levels cover the same range as the CEFR – for vocabulary each token is categorised to a level and counted in the histogram. As well as A1–C2 there is also an unknown (UNK) category. The EVP level for a word is dependent on the word-class and word-sense of the usage. To ascertain word-class (noun, verb *etc.*) we use the *spaCy* *part-of-speech* tagger (Montani et al., 2023) and we select from the EVP the CEFR level for that class. If no such word-class is recorded, we assume the lowest CEFR level for that word as in Tyen et al. (2022). The EVP look-up uses both US and UK English gazetteers. We do not attempt to disambiguate word senses: a word-sense disambiguation model could be trialed in future work.

4.4.5 Essay statistics

Included in the feature set is the number of tokens in the essay (known to correlate with proficiency level) and the revision number (as a raw count).

4.4.6 Profile averages (tCEFR, tGER, tVOCAB)

To help provide a central value against which to measure the offset of grammatical and proficiency scores, we optionally include the mean values for CEFR level, GER and VOCAB for each task.

4.4.7 Avg. Error Deltas (dGER.LT, dCEFR)

These values measure average change in uGER.LT and uCEFR level per submission for each task t .

4.5 Feature and hyperparameter search

The features given in Section 4.4 are filtered to select a subset with high performance. We use *Bayesian optimisation* (Pelikan, 2005) with five-fold cross validation. Specifically, features are selected, and applied uniformly across all five folds. The mean of the validation losses is used in the selection optimisation step. This prevents coincidental outlier scores on particular folds from dominating the tuning process. Hyperparameter search was carried out as part of the same process. Hyperparameters include dropout rate $\in [10^{-10}, 1]$

for each dense layer, number of dense layers on each branch $\in [0, 5]$, width of dense hidden layers $\in [5, 200]$, and choice of activation functions $\in [\text{sigmoid}, \text{linear}, \text{ReLU}]$. In practice, no hidden layers are needed to get good results, and dropout rates are best when small ($< 10^{-3}$).

4.6 Evaluation

We simultaneously predict three targets (the expert assigned values of CEFR level, GER and VOCAB). This section outlines the evaluation for each. For more information on target dimensions and types see A.5.

4.6.1 Evaluation of CEFR level predictions

For the evaluation of CEFR level prediction, we use *Root Mean Squared Error* (RMSE) to provide a measure that can be compared directly to the CEFR. This is calculated globally with equal weighting across all data points (Pelánek, 2018).

4.6.2 Evaluation of GER predictions

We can compare true and predicted GER vector values $\mathbf{g}$ and $\hat{\mathbf{g}}$ using *cosine proximity*:

$$\text{CosProx}(\mathbf{g}, \hat{\mathbf{g}}) = \frac{\sum_k g_k^2 \hat{g}_k^2}{\sqrt{\sum_k g_k^2} \sqrt{\sum_k \hat{g}_k^2}}$$

For positive valued vectors like GER profiles, cosine proximity ranges from $[0, 1]$ with 1 being a perfect match. The use of cosine proximity is based on Zaidi et al. (2019) and it measures the directional similarity of the vectors (*i.e.*, the relative distribution of the magnitudes of components), but it does not take into account vector magnitude. We use *Root Mean Squared Error* (RMSE) as a secondary measure, to check that rates are at the correct levels as well as being distributed in the same way.

4.6.3 Evaluation of vocabulary distributions

Vocabulary distributions are empirical frequency distributions, so our interest is in how similar the histogram is as a whole. We use *Earth Mover Distance* (EMD). For histogram vectors with true value $\mathbf{h}$ and prediction $\hat{\mathbf{h}}$ we have:

$$\begin{aligned} \text{EMD}_0 &= 0 \\ \text{EMD}_{i+1} &= h_i + \text{EMD}_i - \hat{h}_i \\ \text{EMD}(\mathbf{h}, \hat{\mathbf{h}}) &= \sum_{c=1}^{|C|} |\text{EMD}_c| \end{aligned}$$

where there are C classes in each histogram. $\text{EMD} \in [0, 1]$ where 0 is a perfect match.

4.7 Testing

Testing is carried out with a separate held-out test set for each of the five validation folds. In each case, data is partitioned by student, so any student in the test set will not have appeared in either the training or validation data. This ensures that we are measuring generalisable performance and not just re-using samples that have been encountered during training.

5 Results

Table 1 summarises the results of this work, subdivided into the three tasks; for Task 1 we also provide single-feature scoring and best-performing combinations. We also include baseline scores for Zaidi et al. (2019) and the results of our attempts to reproduce this approach.

The first line for Task 1 shows that the *Write & Improve* automarker predicts the expert CEFR level with mean RMSE of 1.403 (where a unit is half a CEFR level). The LT automarker creates GER vectors that are 0.363 cosine proximal to the expert versions. Vocabulary extractions are good predictors, with a mean EMD of 0.012 (out of 1.0) from the expert-corrected case.

We consider three categories (user-essay features comprising Document Basics and Auto Profiles; and Prompt Features) on their own and in combinations. The listing in Table 1 represents a breakdown of the best combinations found by feature filtering.

Some features are clearly more promising than others. In the *Document Basics*, the best try flag achieves the best CEFR prediction with 1.742 RMSE and the *deltas* achieve the best cosine proximity (0.644). For vocabulary distribution the EMD scores are good (0.034-0.035) but vocabulary features are needed to improve this further.

In the *Auto Profile Features* category, uCEFR is the strongest single feature for the CEFR target with 1.050 RMSE, but it is otherwise unremarkable. uVOCAB is the only good predictor of the final vocabulary distribution, dropping the EMD from at best 0.034 to 0.012.

In the *Prompt Features*, we see that tCEFR and tGER have the greatest influence on their respective targets: if we include tCEFR, CEFR RMSE drops to 1.084, so the mean CEFR is a strong indicator as we might expect. Similarly, for tGER alone, GER cos. prox. is good at 0.650 but for other targets it is less strong. The *deltas* alone look promising but are less effective in combination with other features.

	CEFR level	GER	VOCAB
Task 1: Improving profiles	RMSE↓	Cos.Prox.↑	EMD↓
Automarker performance	1.403	0.363	0.007
(Document Basics)			
best.try	1.742	0.643	0.007
n.tokens	1.784	0.640	0.007
version	1.910	0.640	0.007
deltas	1.852	<u>0.644</u>	0.007
best.try+n.tokens+version	<u>1.103</u>	0.623	<u>0.007</u>
best.try+n.tokens+version+deltas	1.106	0.585	0.008
(Auto Profile Features)			
uCEFR	1.050	0.622	0.007
uGER	1.707	0.661	0.007
uGER.LT	1.708	0.657	0.007
uVOCAB	1.297	0.648	0.007
All auto profile features (uALL)	<u>0.932</u>	<u>0.663</u>	<u>0.007</u>
(Prompt Features)			
tCEFR	1.084	0.626	0.007
tGER	1.595	0.650	0.007
tVOCAB	1.267	0.649	0.007
All prompt features (tALL)	<u>1.076</u>	<u>0.650</u>	<u>0.007</u>
(Replicating Zaidi et al., 2019)			
Prev. Baseline	1.383	-	-
Prev. Best	1.093	0.426	-
Closest (uGER+best)	1.045	0.639	0.007
uALL+tALL (excl. VOCAB)	0.848	0.664	0.007
uALL+tALL+best.try (excl. VOCAB)	<u>0.838</u>	<u>0.665</u>	<u>0.007</u>
(Best Overall Combinations)			
uALL+tALL	0.817	0.670	0.007
uALL+tALL+n.tokens	0.816	0.664	0.007
uALL+tALL+deltas	0.817	0.664	0.007
uALL+tALL+best.try	0.809	0.670	0.007
Task 2: Forward prediction			
Automarker performance	1.426	0.339	0.006
Best from Task 1	0.874	0.659	0.006
Task 3: Generalisation			
Automarker performance	1.913	0.283	0.007
Best from Task 1	0.963	0.649	0.007

Table 1: Results for predicting human expert profile scores from auto profiles. Best features or combinations for each subclass are underlined, best combinations for each Task are in **bold**. ↑=higher is better; ↓=lower is better.

The most performant combination is all *Auto Profile* and *Prompt* features, with the best.try metric. This results in CEFR RMSE of 0.809, GER cos. prox. 0.670 and VOCAB EMD of 0.011.

For comparison with previous work we include a baseline and best score from Zaidi et al. (2019), which is the closest precedent to our Task 1. No baseline grammar scores were given and vocabulary was not included in that paper. In Zaidi et al.

the highest performing configuration used just GER profile and best.try metric – with these features we achieve CEFR RMSE of 1.045 and GER cos. prox. of 0.639, both better than the earlier work. With our best configurations, but excluding VOCAB entirely (as per Zaidi et al.) we do better still, with CEFR RMSE of 0.838 and GER cos. prox. of 0.665.

In Task 2, we attempt to predict the final expert profile, using auto profiles from earlier drafts. Here,

using the best configuration from Task 1, we see CEFR RMSE of 0.874 and GER cos. prox. of 0.659. VOCAB EMD is 0.014. These scores are slightly worse than for Task 1, although still beating the Task 1 automarkers.

In Task 3, we attempt to predict the expert profile for the final draft of the *next* series that the student will undertake. In this task we see CEFR RMSE of 0.963 and GER cos. prox. of 0.649. Vocabulary scores, although low in the full range of EMD, are half as good as in other tasks, with EMD 0.028.

6 Discussion

The multi-task learning configuration used in this work is currently the best recorded in this class of task and data. In Zaidi et al. (2019) a larger, private data set is used but the best performance (using an ERRANTised GER vector and the best-essay flags to target CEFR level) is weaker in terms of both CEFR level prediction and GER cosine proximity (1.093 to 0.809 RMSE and 0.426 to 0.669 cos. prox. respectively).

The implication is that a richer target is just as important as having richer input features (which are tested without leading to any improvement in Zaidi et al.). The 84% increase in GER cosine proximity is a key result, but consistent GER RMSE of 0.006-0.007 is also better than we might expect, thanks to the choice of loss function. Retrained with negative cosine proximity as the loss, our best system returns similar cosine proximity but GER RMSE 0.332, much worse than with Euclidean loss.

Outside of prediction, we can inspect the internal representations output by the interior branches of the network. For instance we can extract and concatenate the CEFR, GER and VOCAB activations just for the *user* branches. This is shown for 200 randomly selected students (at submission of final essay) projected to 2D using MDS (Kruskal, 1964) in Figure 5. Colours are (a) expert CEFR and (b) the user's choice of prompt difficulty. Neither of these is used in training, but banded clustering is evident for both, which suggests these representations could be used to apply adaptive pedagogical policies, such as changing a student's prompt group when their characteristics do not fit their current grouping.

7 Visualiser

We make the data from this work available in a visualisation tool⁷ (See figure 6). For each student the tool presents personalised profile data: (1) Student learning statistics, including average human and automarker CEFR level, ranks in cohort, average number of drafts, and rate of change of CEFR and average GER per draft. (2) CEFR level time series. (3) Average grammatical error rate (GER) time series. (4) Grammatical error rate (GER) summary. (5) Vocabulary distribution (VOCAB) summary. In the time series and grammar charts, the gold line shows our predicted score, teal shows the human expert scores and cyan shows the automarker scores.

8 Limitations and Future Work

The technique described in this work is simple and fast to train, but it is *Markovian* and assumes that the effect of history has been incorporated into the features seen at each timestep. While it may be reasonable to model skill as cumulative in this way, it may be better to include history explicitly using a method such as Knowledge Tracing (Corbett and Anderson, 1994; Piech et al., 2015). This is scheduled for future work. Another limitation is that this study has been restricted to a single data-set and platform but this is hard to rectify since there is a scarcity of such corpora.

9 Conclusion

We introduce interpretable multi-component vectors, *writing profiles*, that describe a learner's writing in terms of CEFR level, and vocabulary usage alongside their grammatical errors. Using a neural model, we effectively align automatically generated writing profiles with high-quality human-expert profiles, enhancing AES performance. Further, the neural model is shown to generalise across essay drafts and future tasks; paving the way for future work in longitudinal curriculum planning. Lastly, we make writing profile data explorable within our longitudinal visualisation tool.

Acknowledgements

This paper reports on research supported by *Cambridge University Press & Assessment*.

⁷https://github.com/rjm49/bea26_visualiser

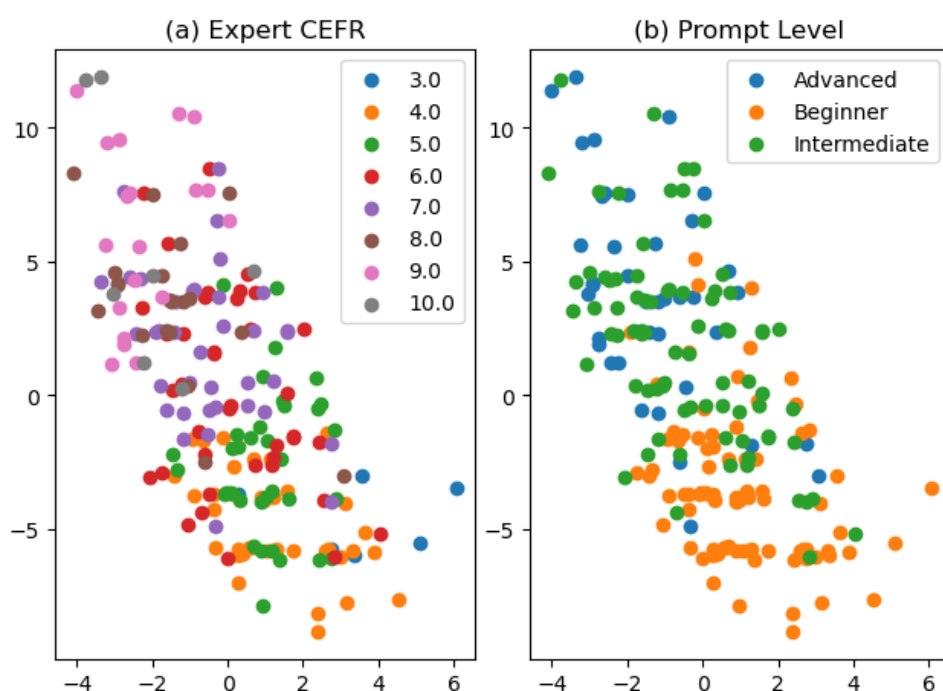


Figure 5: Representations of students from the test set, concatenated from the branches of the neural network associated with the user only. These are projected into 2D by MDS, and coloured by (a) expert CEFR and (b) prompt (difficulty) levels. These colour groupings are not known to the system, so the scatters reveal that this structure has been learned independently. This may be helpful in supporting pedagogical adaptivity such as deciding which prompt group a student should actually belong to.

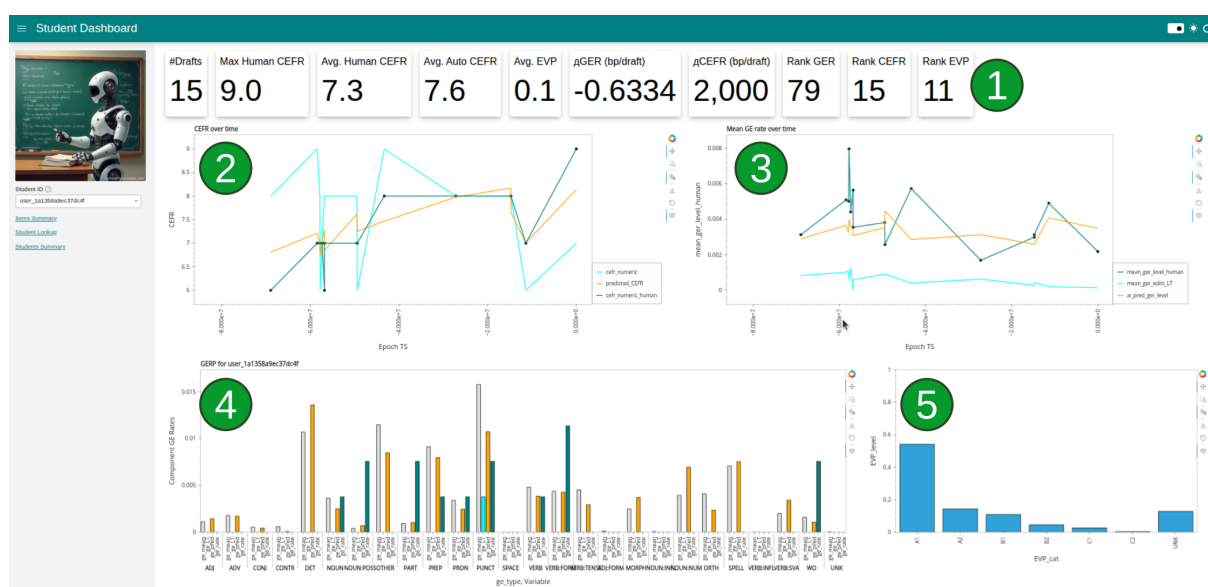


Figure 6: Visualiser Screenshot - student detail view. For each student the tool presents: (1) Student learning statistics. (2) Student CEFR level time series. (3) Student average grammatical error rates time series. (4) Grammatical error summary. (5) Vocabulary usage summary. In the time series and grammar summary charts, the gold line represents our predicted score, dark teal is expert score, cyan is automarker score.

References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, and 21 others. 2015. [TensorFlow: Large-scale machine learning on heterogeneous systems](#). Software available from tensorflow.org.
- Øistein E Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. [Developing and testing a self-assessment and tutoring system](#). *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 32–41.
- John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. [Cognitive tutors: Lessons learned](#). *The Journal of the Learning Sciences*, 4(2):167–207.
- John S. Bridle. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neural Networks for Signal Processing*, pages 227–236. MIT Press.
- 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.
- Annette Capel. 2015. The English Vocabulary Profile. In J. Harrison and F. Barker, editors, *English Profile in Practice*. Cambridge: Cambridge University Press.
- Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis—a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, pages 164–175. Springer.
- François Chollet and 1 others. 2015. Keras. <https://keras.io>.
- 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.
- Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2012. [The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning](#). *Cognitive Science*, 36(5):757–798.
- Joseph B Kruskal. 1964. [Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis](#). *Psychometrika*, 29(1):1–27.
- Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.
- Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. [Can large language models automatically score proficiency of written essays?](#) In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.
- 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.
- Marcin Miłkowski. 2010. [Developing an open-source, rule-based proofreading tool](#). *Software: Practice and Experience*, 40(7):543–566.
- 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](#).
- Diane Nicholls, Andrew Caines, and Paula Buttery. 2024. [The Write & Improve Corpus 2024: Error-annotated and CEFR-labelled essays by learners of English](#). Cambridge University Press & Assessment.
- Brian North. 2005. The CEFR levels and descriptor scales. In Lynda Taylor and Cyril Weir, editors, *Multilingualism and assessment: Achieving transparency, assuring quality, sustaining diversity. Proceedings of the ALTE Berlin Conference*. Cambridge: Cambridge University Press.
- 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.
- Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. [Performance factors analysis—a new alternative to knowledge tracing](#). In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*.
- 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.

- Radek Pelánek. 2018. [The details matter: methodological nuances in the evaluation of student models](#). *User Modeling and User-Adapted Interaction*, 28(3):207–235.
- Martin Pelikan. 2005. [Bayesian optimization algorithm](#). In *Hierarchical Bayesian optimization algorithm: toward a new generation of evolutionary algorithms*, pages 31–48. Berlin, Heidelberg: Springer.
- Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. [Deep knowledge tracing](#).
- Georg Rasch. 1980. *Probabilistic models for some intelligence and attainment tests*. Chicago: University of Chicago Press.
- 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.
- Helen Yannakoudakis, Øistein E Andersen, Ardeshtir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. [Developing an automated writing placement system for ESL learners](#). *Applied Measurement in Education*, 31(3):251–267.
- Ahmed H Zaidi, Andrew Caines, Christopher Davis, Russell Moore, Paula Buttery, and Andrew Rice. 2019. [Accurate modelling of language learning tasks and students using representations of grammatical proficiency](#). In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*.

A Appendix

A.1 CEFR Level Mapping

Level	Numeric
pre-A1	0
A1	1-2
A2	3-4
B1	5-6
B2	7-8
C1	9-10
C2	11-12

Table 2: CEFR Level Mappings: note that each CEFR level is divided into two sublevels, e.g. A1 and A1+

A.2 ERRANT Error Categories

Code	Description
ADJ	Adjective
ADJ:FORM	Adjective form
ADV	Adverb
CONJ	Conjunction
CONTR	Contraction
DET	Determiner
NOUN	Noun
NOUN:POSS	Possessive noun
MORPH	Morphology
NOUN:INFL	Noun inflection
NOUN:NUM	Noun number
ORTH	Orthography
OTHER	Other
PART	Particle
PREP	Preposition
PRON	Pronoun
PUNCT	Punctuation
SPACE	Whitespace
SPELL	Spelling
VERB	Verb
VERB:FORM	Verb form
VERB:INFL	Verb inflection
VERB:SVA	Verb subject-verb-agreement
VERB:TENSE	Verb tense
WO	Word order
UNK	Unknown

Table 3: ERRANT codes. ERRANT marks correction as belonging to one of these categories with an additional (R)eplaced (M)issing or (U)nnecessary marker to describe whether edit has changed, added to, or deleted from, the original text. We drop the R/M/U markers and just keep the categories listed.

A.3 Language Tool Error Categories

Language Tool Category	Description
CASING	Incorrect upper and lower casing.
COLLOCATIONS	Incorrect word combinations.
COMPOUNDING	Hyphenation and compound word errors.
CONFUSED WORDS	Commonly confused words.
GRAMMAR	Syntactic errors.
MISC	Other errors (unspecified).
MULTITOKEN SPELLING	Words incorrectly split or merged.
NONSTANDARD PHRASES	Slang and inappropriate colloquial forms.
PUNCTUATION	Missing or incorrect punctuation marks.
REDUNDANCY	Unnecessary repetition or wordiness.
REPETITIONS	Immediate or local duplication.
STYLE	Problems with register, readability and verbosity.
TYPOGRAPHY	Errors in spacing, quotes and other symbology.
TYPOS	Misspellings.

Table 4: *Language Tool* error categories, with descriptors.

A.4 Input Features

Input Name	Width, Type
uGER	26, float
uGER.LT	14, float
uCEFR	1, integer
uVOCAB	7, float
best.try	2, bit
n.tokens	1, integer
version	1, integer
dGER.LT	14, float
dCEFR	1, float
tGER	26, float
tCEFR	1, float
tVOCAB	7, float

Table 5: Dimensions and types for the input features used in this work.

A.5 Prediction Targets

Target Name	Width, Type
GER	26, float
CEFR	1, integer
VOCAB	7, float

Table 6: Dimensions and types for the prediction targets used in this work.

Policy-Sensitive Fairness Evaluation in Automated Scoring of Clinical Communication

Saed Rezayi¹, Le An Ha², Victoria Yaneva¹
Polina Harik¹, Janet Mee¹, Jason Snyder¹

¹NBME ²Ho Chi Minh City University of Foreign Languages

(srezayidemne,vyaneva,pharik,jmee,jsnyder)@nbme.org anh1@hufliit.edu.vn

Abstract

This study examines automated scoring fairness in a formative assessment context: the automated evaluation of medical students' communication skills. Building on the premise that definitions of fairness are value-dependent, we investigate how conclusions about group differences may vary under different weighting schemes for false positives (FPs) and false negatives (FNs). Results show that when errors are treated symmetrically, no statistically significant differences are observed across demographic groups based on race or gender. This pattern remains stable when error weights are varied, with no consistent or robust disparities emerging. A small number of isolated differences appear under moderate FN weighting. Overall, the findings suggest that fairness conclusions in this setting are relatively robust to variations in error weighting. At the same time, the study highlights the importance of making value assumptions explicit when evaluating automated scoring systems, particularly in formative contexts where error trade-offs carry pedagogical implications for feedback, learner engagement, and educational equity.

1 Introduction

Automated scoring systems are increasingly used in educational settings to evaluate constructed responses at scale (Flores et al., 2025). They can reduce scoring burden and support timely feedback, but they also raise an important question: does the system behave differently across demographic groups in ways that reflect construct-irrelevant factors? This concern is especially important in cases where scores may affect learning opportunities, consequential decisions, and perceptions of competence.

Fairness is an inherent concern in algorithmic classification, and a consistent finding across the literature is that fairness is not a single, unified concept. Notable papers such as “On the

(im)possibility of fairness” (Friedler et al., 2021) showcase how different notions of fairness reflect different underlying value systems, and no single classification system can satisfy all of them simultaneously.

This challenge is equally present in research on automated scoring in education (Johnson and McCaffrey, 2023). In this field, a fair test is one in which differences in scores among test-takers reflect only variations in the skills defined by the construct. Systematic score differences between non-random groups that arise from factors unrelated to the construct (i.e., construct-irrelevant factors) may signal potential unfairness in the assessment (Litman et al., 2021; Schaller et al., 2024; Andersen et al., 2025). To illustrate these competing perspectives, Loukina et al. (2019) distinguish three aspects of fairness: (1) whether scores are equally accurate across groups, (2) whether scores systematically differ from human scores for certain groups, and (3) whether such discrepancies persist when individuals have comparable ability. Empirical findings corroborate that fully satisfying all criteria simultaneously is rarely feasible.

While prior work on fairness in automated scoring has primarily focused on *summative assessment*, particularly for essay scoring and short-answer grading (Johnson et al., 2022), this paper examines fairness in *formative assessment* and in a different domain: the automated evaluation of medical students' communication skills with patients.

When discussing fairness in automated scoring in learning tools, there could be different value systems that govern the definition and subsequent evaluation of fairness. In the present setting, our automated scoring system makes binary decisions about whether specific behaviors are present in a learner response. In this setting, one key consideration is the relative importance of false positives (FPs) versus false negatives (FNs), as discussed in several key studies (Dey et al., 2024; Ernst et al.,

2025). In formative assessment, these trade-offs have pedagogical consequences: FNs may undermine learner confidence, while FPs can obscure learning gaps. Some value systems treat errors symmetrically, prioritizing overall accuracy; others emphasize minimizing FPs, ensuring positive evaluations reflect demonstrated competence; still others prioritize minimizing FNs to avoid discouraging learners or overlooking emerging skills. For example, Li et al. (2023) examine the effects of FPs (incorrect answers marked correct) and FNs (correct answers marked incorrect) in an AI autograder for short-answer tasks. The findings show that FPs can harm learning, largely because students fail to detect them and disengage after receiving positive feedback. FNs have more variable effects, with evidence that deeper engagement can mitigate their impact. In another study, Dey et al. (2024) find that the odds of addressing a FP feedback was 99% lower than addressing a FN feedback, representing significant missed opportunities for revision and learning. Thus, evaluation criteria reflect not only technical choices but also assumptions about which errors are most consequential for learning.

This study examines how fairness across demographic groups may vary under these different error-weighting approaches. If one type of error disproportionately affects certain learners, feedback may be misleading, impacting motivation and learning outcomes. Understanding these dynamics is crucial for designing automated scoring systems that are both accurate and educationally equitable. The paper makes the following original contributions:

- Examines fairness in automated scoring of medical students' communication skills in a formative context.
- Investigates the impact of error-weighting on fairness across demographic groups.
- Highlights the pedagogical implications of error asymmetries for feedback quality, learner motivation, and educational equity.

2 Data

Context: The context for this study is the Communication Learning Assessment (CLA) learning tool¹. CLA is a formative, digital tool designed to support medical students in developing communication skills for effective interaction with patients. CLA presents learners with video-based patient

Group	Sample size (<i>n</i>)	PCB rate (%)
<i>Gender</i>		
Female	837	61.4
Male	522	60.7
Non-binary	16	56.3
Other gender	8	87.5
<i>race</i>		
White/Caucasian	799	60.8
Asian	322	59.9
African American	150	63.3
Multiracial	56	58.9

Table 1: Dataset summary: subgroup sizes and annotation PCB prevalence ($N = 1,383$).

scenarios that describe a clinical case. Each case ends with a question or statement from the patient. The learner can then record a verbal response of up to two minutes addressing the patient, where they strive to exemplify appropriate communication skills. The three main categories of skills practiced via these cases include 1) responding to emotions, 2) providing information, and 3) fostering the relationship. Within each of these categories, the learner is expected to demonstrate appropriate communication behaviors within the specific clinical case they are given, which are called *patient-centered behaviors* (PCBs). For example, when responding to emotions in a case about a patient sharing discouragement that their weight loss journey has plateaued, an appropriate PCB might be “*I am sorry to hear this, it must be truly frustrating after all these efforts*”. Notably, these PCBs can vary widely among learners and cases, which makes PCB annotation and, subsequently, detection, a non-trivial task. In CLA, automated detection of case-specific PCBs is used to generate scores and those scores can form a basis of individualized feedback for learners.

Learners: The learners in our dataset were 3rd and 4th year US medical students who had successfully passed the USMLE® Step 1 exam² and used the CLA tool for practicing their communication skills. The dataset includes self-reported gender and race for each learner.

Annotation: Responses from 223 learners across 8 clinical cases were annotated for PCBs, although not every learner responded to every case. Annotation was performed by a team of four NBME staff members trained in clinical communication assessment. Annotations were guided by a detailed rubric (see Appendix A), capturing

¹<https://www.nbme.org/cla>

²<https://www.usmle.org/step-exams/step-1>

PCBs essential for effective physician-patient interactions, such as *acknowledgment of patient concerns*, *provision of clear explanations*, *demonstration of empathy*, and *reinforcement of positive behaviors*. This rubric included 25 unique PCBs, each associated exclusively with one of the 8 clinical cases, with each case containing between 1 and 4 distinct PCBs. Annotators were instructed to precisely identify text spans corresponding to each PCB by providing exact character-level indices. Negative samples for each PCB were systematically derived by listing all learner responses from the same clinical case that were not annotated as reflecting that specific PCB.

Annotators scored de-identified transcripts, were blind to learner demographics, and received explicit bias training. We therefore use human annotations as the reference standard for PCB presence, while recognizing that this does not rule out all sources of annotation bias.

Automated scoring system: The automated scoring of PCBs was performed by training ACTA (Rezayi et al., 2025) – an encoder-based transformer scoring model built on DeBERTa used operationally within CLA – on a cross-validated set of learner responses. Average binary F1 score for each PCB was 0.92 (min = 0.86, max = 0.97). Fairness was evaluated using out-of-fold predictions from 5-fold cross-validation on the annotated dataset.

Dataset: The resulting dataset consists of 1,383 records. Each record includes self-reported learner demographics, human PCB annotations, and ACTA predictions by learner response (encounter) and PCB type. Each prediction is assigned TP/TN/FP/FN labels after a careful error analysis by the annotation team. Each record includes the clinical communication text, a PCB label, and identifiers (case_id, encounter_id).

Table 1 reports subgroup sizes and human annotation prevalence (PCB rate). The largest groups are Female and White/Caucasian learners. Importantly, annotation PCB prevalence is nearly constant across high-volume groups (approximately 59–63%). Smaller groups (e.g., non-binary and other gender identities) have limited sample size and are therefore reported descriptively.

Looking at individual clinical cases (Table 4 in Appendix B), some cases show differences in PCB prevalence rates, but these differences are not statistically significant. In case 180, the PCB prevalence rate for the White/Caucasian group is around 10%

lower than for the other two high-volume groups, while case 192 shows the reverse pattern. This suggests that differences in individual cases may reflect factors other than annotation bias.

3 Methodology

We use prediction scores from the operationally deployed ACTA model. ACTA produces binary PCB-level decisions for each clinical encounter (learner-case interaction), and a single encounter may contain multiple PCBs. Because these decisions are clustered within encounter, we use the encounter as the unit of analysis rather than treating PCB-level observations as independent.

For encounter e we define a policy-sensitive error score by summing weighted false negatives and false positives across all PCBs associated with the encounter’s case:

$$L(e) = \sum_{c \in \mathcal{C}_{v(e)}} [w^{\text{FN}} \text{fn}_{e,c} + w^{\text{FP}} \text{fp}_{e,c}] \quad (1)$$

Here $\text{fn}_{e,c}$ and $\text{fp}_{e,c}$ are binary indicators of FN and FP errors for PCB c , and w^{FN} and w^{FP} control their relative importance. These weights are part of the evaluation, not the model.

To compare demographic groups, we compare case-adjusted disparity in mean encounter-level error. For groups a and a' ,

$$\Delta(a, a') = \sum_v \pi(v) \left| \mathbb{E}[L(e) \mid A=a, V=v] - \mathbb{E}[L(e) \mid A=a', V=v] \right| \quad (2)$$

where V is the vignette and $\pi(v)$ is the proportion of encounters in the full dataset that belong to vignette v , where $\sum_v \pi(v) = 1$.

4 Experiments

We evaluate ACTA with two views of fairness. First, we report PCB-level metrics by demographic group: recall, precision, F1, and positive prediction rate. Second, we compute the encounter-level disparity measure $\Delta(a, a')$ under multiple FN/FP weight ratios to test whether fairness conclusions are stable to different error priorities. Uncertainty is estimated with 1,000 bootstrap resamples at the encounter level, stratified by case so that case composition is preserved.

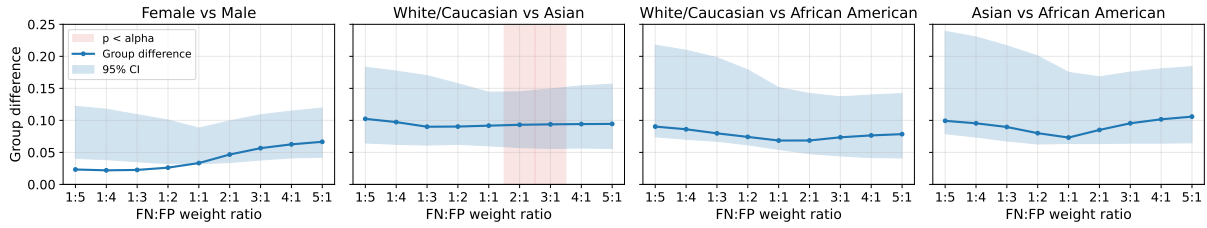


Figure 1: Absolute encounter-level disparity across different FN and FP weightings. The x-axis shows the FN/FP weight ratio, and the y-axis shows the absolute disparity estimate for each pairwise comparison.

Attribute	Group	Recall	Prec.	F1	PPR
Gender	Female	0.963	0.926	0.944	0.661
Gender	Male	0.960	0.908	0.933	0.633
Race	White/Caucasian	0.967	0.928	0.947	0.637
Race	Asian	0.931	0.915	0.923	0.637
Race	African American	0.979	0.906	0.941	0.677

Table 2: PCB-level performance metrics for the high-volume demographic groups in CLA.

Table 2 shows that the PCB-level metrics are broadly similar across the high-volume groups, with only small gender differences. Race shows somewhat more descriptive variation, especially in recall for comparisons involving Asian students, but none of the pairwise group differences is statistically significant after Bonferroni correction. Table 3 shows that the same general pattern holds for the encounter-level disparity measure under equal weighting of FNs and FPs: none of the pairwise comparisons is significant after correction, although the largest baseline disparity is observed for the White-Asian comparison.

Figure 1 shows that this baseline view is not fully stable across error-priority settings. As the relative weight assigned to FNs increases, absolute encounter-level disparity increases for all four pairwise comparisons. This indicates that group differences become larger when missed credit is treated as more costly than undeserved credit. At the same time, the comparisons do not behave identically: the White-Asian disparity is the only one that reaches significance anywhere in the sweep, doing so at the 2:1 and 3:1 FN:FP settings, while the remaining comparisons remain below the pairwise significance threshold. The White/Caucasian–Asian result seems to be mainly about false negatives. In Table 2, the Asian subgroup has lower recall than the White/Caucasian subgroup, while precision is more similar. This means the model misses more true PCBs for Asian learners, but it does not show an equally large difference in false positives. Because of that, the group difference becomes larger when false negatives are

Pair tested	$\Delta(a, a')$	95% CI	p-value
Female vs Male	0.016	[0.000, 0.108]	0.723
White/Caucasian vs Asian	0.090	[0.015, 0.182]	0.082
White/Caucasian vs African American	0.021	[0.000, 0.160]	0.748
Asian vs African American	0.055	[0.000, 0.205]	0.478

Table 3: Encounter-level disparity estimates under equal weighting of FNs and FPs. None of the pairwise comparisons were statistically significant.

given more weight. We cannot say from this analysis why those missed detections happen. For this reason, we interpret the result cautiously as a local sensitivity to FN weighting, not as evidence of a broad or stable demographic disparity.

5 Discussion

The main finding from this study is that when errors are treated symmetrically—i.e., when FPs and FNs are assigned equal weight—the empirical results indicate no statistically significant differences across demographic groups. This suggests that, under a value framework in which both types of errors are considered equally consequential, the scoring system performs comparably across groups. This finding is particularly relevant because many intended uses of these scores assume such symmetry in error costs.

Next, we investigate patterns when weights on different types of errors are disproportionately increased. When increasing the weights on FPs up to fivefold, no statistically significant differences between groups are observed, suggesting that under a value framework that emphasizes high accuracy, the system performs equitably. A similar pattern holds when increasing the weights of FNs, with one exception: statistically significant differences emerge when FNs are assigned two- or threefold weights in the White–Asian comparison. We emphasize that we do not currently have a clear theoretical explanation for this pattern, and thus interpret it with caution, particularly in light of the fact that the effect disappears at higher weights (e.g., four- or fivefold). One possible account is that it

may be an example of Type I error resulting from multiple comparisons, despite the application of Bonferroni correction. It may also reflect a combination of sampling variability, subgroup imbalances, and underlying distributional differences in scores that render certain comparisons more sensitive to moderate changes in FN weighting. In particular, such reweighting may disproportionately affect borderline cases, producing localized effects. Overall, these findings point to a complex interaction between data characteristics and metric design, with a likely minimal practical effect.

6 Limitations and Ethical Considerations

Several limitations should be considered when interpreting the findings of this work.

First, the dataset is relatively limited in size, particularly for certain subgroup comparisons. This constrains statistical power and reduces the ability to detect small but potentially meaningful differences across demographic groups. This limitation is especially relevant for intersectional analyses, where subgroup counts become even smaller.

Second, the analysis is conducted within the specific context of the ACTA scoring system and the CLA learning tool. While these provide a valuable and realistic testbed, they represent a particular type of automated scoring application with its own design choices and use cases. Consequently, the findings may not generalize to other automated scoring settings, such as those involving different constructs, item formats, model architectures, or deployment contexts. Replication across diverse systems and domains is necessary to establish the broader applicability of these results.

Third, the weighting schemes used in this study are intentionally simple, focusing on systematically varying the relative importance of FPs and FNs. While this approach is useful for conducting sensitivity analyses and illustrating how fairness conclusions shift under different value assumptions, it does not fully capture the complexity of real-world assessment policies. In practice, decisions about error costs may depend on a wider range of factors, including the stakes of the assessment, the distribution of scores, downstream decision-making processes, and stakeholder priorities. More nuanced or context-specific cost functions could lead to different patterns of results.

In terms of ethical considerations, institutions deploying automated scoring systems have a respon-

sibility to examine disaggregated error patterns, assess the potential harms associated with different error types, and ensure that mechanisms exist for identifying and correcting erroneous scores. Such analyses help ensure that automated scoring systems do not introduce or amplify inequities through the unequal distribution of scoring errors.

A key consideration is that while most investigations of group differences treat FP and FN errors as equally important, the intended use of an assessment should guide the relative weight of these errors. The consequences of awarding credit without merit (FP) or withholding credit when it's deserved (FN) can differ substantially in low vs. high stakes and formative vs. summative assessments. For example, in a high-stakes context like a medical licensure exam, incorrectly passing an unqualified physician (FP) can potentially result in risk to real patients. In this case, minimizing FP errors might be prioritized. In contrast, for low-stakes formative tools, such as CLA, failing to award credit when it's deserved (FN) can affect a learner's confidence and motivation, ultimately undermining the main purpose of the learning tool.

These differences underscore that error weighting is fundamentally a value-laden decision, tied to the goals and consequences of the assessment. Treating FP and FN errors symmetrically may be appropriate in some cases, but it should not be assumed as a default. Instead, evaluation frameworks should explicitly align error trade-offs with the intended use of the assessment, as different choices can lead to materially different conclusions about system performance and fairness across groups.

A remaining source of uncertainty is that both human annotation and automated scoring operate on the same de-identified text. As a result, construct-irrelevant linguistic variation correlated with demographic group could affect both human labels and model predictions. Potential sources include subgroup differences in lexical, grammatical, or discourse patterns, as well as uneven subgroup representation in model training.

References

- Nico Andersen, Julia Mang, Frank Goldhammer, and Fabian Zehner. 2025. Algorithmic fairness in automatic short answer scoring. *International Journal of Artificial Intelligence in Education*, pages 1–38.
- Indrani Dey, Dana Gnesdilow, Rebecca Passonneau, and Sadhana Puntambekar. 2024. Potential pitfalls

of false positives. In *International Conference on Artificial Intelligence in Education*, pages 469–476. Springer.

Helen M Ernst, Anja Prinz-Weiß, Jörg Wittwer, and Thamar Voss. 2025. Discrepancy between performance and feedback affects mathematics student teachers’ self-efficacy but not their self-assessment accuracy. *Frontiers in Psychology*, 15:1391093.

Gerardo Flores, Alyssa Hasegawa Smith, Julia Fukuyama, and Ashia C. Wilson. 2025. [Aligning evaluation with clinical priorities: Calibration, label shift, and error costs](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2021. The (im) possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM*, 64(4):136–143.

Matthew S Johnson, Xiang Liu, and Daniel F McCaffrey. 2022. Psychometric methods to evaluate measurement and algorithmic bias in automated scoring. *Journal of Educational Measurement*, 59(3):338–361.

Matthew S Johnson and Daniel F McCaffrey. 2023. Evaluating fairness of automated scoring in educational measurement. *Advancing natural language processing in educational assessment*, 142.

Tiffany Wenting Li, Silas Hsu, Max Fowler, Zhilin Zhang, Craig Zilles, and Karrie Karahalios. 2023. Am I wrong, or is the Autograder wrong? Effects of AI grading mistakes on learning. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*, pages 159–176.

Diane Litman, Haoran Zhang, Richard Correnti, Lindsay Clare Matsumura, and Elaine 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. Springer.

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.

Saed Rezayi, LA Ha, Y Zhou, et al. 2025. Automated scoring of communication skills in physician–patient interaction: Balancing performance and scalability. In *Proc 20th BEA Workshop*.

Nils-Jonathan Schaller, Yuning Ding, Andrea Horbach, Jennifer Meyer, and Thorben Jansen. 2024. Fairness in automated essay scoring: A comparative analysis of algorithms on german learner essays from secondary education. In *Proceedings of the 19th workshop on innovative use of nlp for building educational applications (bea 2024)*, pages 210–221.

A Annotation Guideline

1. **Annotate Complete Components.** Ensure each annotation captures the complete text relevant to the label, even if the sentence is grammatically incomplete.
2. **Annotate all instances of a concept or label.** More variations of a label or concept will better train the automated scoring model.
3. **Include Filler Words Within a Text Span.** Stutters, repetitions, false starts, and filler words such as ‘um,’ ‘ok,’ or ‘like’ should be included only if they are in the middle of a text span. Do not annotate them if they are at the beginning or end.
4. **Do Not Infer Beyond the Text.** Annotate only what is explicitly stated. Avoid making assumptions or interpretations beyond the text.
5. **Maintain Neutrality.** Do not let personal opinions or interpretations influence the annotation. Focus solely on the content and structure. Transcripts are not being scored; they serve as a source of information to train an automated scoring engine.
6. **Follow Label Definitions Strictly.** Always refer to the label definitions provided. If a segment doesn’t clearly fit a label, leave it unannotated or flag it for review.
7. **Parsimony.** Annotate only the minimal number of words necessary to accurately represent the label, avoiding extraneous or redundant text to maintain clarity and consistency across annotations.
8. **Separate Labels for Adjacent Components.** When two adjacent statements or questions both correspond to the same label, annotate each one separately.
9. **Punctuation Does Not Affect Annotation.** Include or exclude it as needed, focusing instead on accurately capturing the relevant words for the label.
10. **Avoid Overlap.** Annotations can sometimes seem to have similar concepts. However, if there is overlap in the annotation, double-check the label definition and/or flag for review, as there should be a distinction and minimal overlap.

case id	174	175	176	177	178	180	182	192
n encounters	54	52	54	59	46	55	58	59
n PCBs	3	1	3	4	3	3	4	4
<i>Number of positive PCB annotations (prevalence rate in %)</i>								
<i>By Gender</i>								
Male	91 (56%)	52 (100%)	80 (49%)	164 (69%)	56 (41%)	99 (60%)	171 (74%)	134 (57%)
Female	48 (52%)	33 (100%)	41 (46%)	98 (72%)	32 (41%)	63 (64%)	109 (72%)	90 (58%)
<i>By Ethnicity</i>								
White/Caucasian	55 (59%)	31 (100%)	39 (45%)	98 (70%)	34 (42%)	54 (55%)	97 (73%)	78 (57%)
Asian	21 (54%)	11 (100%)	27 (56%)	32 (73%)	15 (38%)	22 (67%)	34 (71%)	31 (52%)
African American	8 (44%)	6 (100%)	10 (56%)	26 (72%)	2 (33%)	12 (67%)	24 (75%)	7 (44%)

Table 4: Number of encounters, and PCBs, and positive annotations by case, gender, and ethnicity.

B Data Statistics

Table 4 summarizes the eight cases included in the analysis. The number of encounters is similar across cases, ranging from 46 to 59, while the number of PCBs ranges from 1 to 4. Positive annotation rates vary across cases, but differences by gender and ethnicity within each case are generally small.

Noise Steering for Controlled Text Generation: Improving Diversity and Reading-Level Fidelity in Arabic Educational Story Generation

Haziq Mohammad Khalid
b00100601@aus.edu

Salsabeel Shapsough
sshapsough@aus.edu

Imran Zualkernan
izualkernan@aus.edu

American University of Sharjah

Abstract

Generating diverse, pedagogically valid stories for Arabic early-grade reading assessments requires balancing tight constraints on vocabulary, reading level, and narrative structure against the need to avoid repetitive plots that undermine assessment validity. We investigate *noise steering*, injecting calibrated Gaussian perturbations into the internal representations of transformer models at inference time, as a training-free diversity method evaluated across five small Arabic-centric language models (7–9B parameters). We compare four injection strategies against high-temperature sampling baselines, measuring diversity, quality, constraint adherence, and reading grade level. Residual stream noise consistently improves narrative diversity with minimal quality or constraint cost and preserves early-grade reading level across all Arabic-centric models. Attention entropy noise injection (AENI) stabilizes the otherwise unreliable attention-logit noise while recovering quality. High-temperature sampling inflates reading grade level and causes catastrophic collapse on several models. We find internal representation-level perturbation to be a more suitable diversity strategy than output-level stochasticity for constrained educational content generation.

1 Introduction

The Early Grade Reading Assessment (EGRA) is a widely used framework for evaluating foundational reading skills in young learners (Dubeck and Gove, 2015). A key component involves short narrative passages that must satisfy strict pedagogical constraints: controlled vocabulary, appropriate reading level, clear narrative structure, and cultural neutrality. Manually producing such passages at scale is labor-intensive, making automatic generation an attractive alternative, particularly in Arabic, where rich morphology, orthographic variation, and diglossia introduce additional linguistic complexity (Habash, 2010). Recent work has

shown that large language models can generate fluent Arabic stories aligned with EGRA criteria, but two tensions persist (Abdelghafur et al., 2025; Rai et al., 2024). First, *diversity vs. constraint adherence*: increasing output variety risks violating the tight pedagogical requirements that make EGRA passages valid assessment instruments (Holtzman et al., 2020). Second, practical deployment in under-resourced educational settings favors small language models (SLMs) of 7–9 billion parameters over large frontier models. Yet, these smaller Arabic-centric models are more susceptible to generation instability when diversity is pushed. We address both tensions by investigating *noise steering*: injecting calibrated Gaussian perturbations into the internal representations of transformer models at inference time, without any fine-tuning. We evaluate four injection strategies across five Arabic SLMs and show that residual stream noise and attention entropy noise injection (AENI) consistently improve narrative diversity while preserving reading grade level and constraint adherence, outperforming output-level approaches such as high-temperature sampling that raise diversity at the cost of pedagogical validity.

2 Background

2.1 EGRA Constraints

EGRA-aligned texts are constructed under strict pedagogical and linguistic constraints (Dubeck and Gove, 2015). These include controlled passage length, age-appropriate vocabulary, a clear narrative structure, limited characters, grammatical correctness, and cultural neutrality. These constraints ensure comparability across assessments and support the generation of both literal and inferential comprehension questions. However, manually creating such passages is labor-intensive and requires domain expertise, limiting scalability (Abdelghafur et al., 2025).

2.2 Challenges in Arabic EGRA Generation

Generating EGRA-aligned content in Arabic introduces additional challenges due to the language’s rich morphology, orthographic ambiguity, and diglossia between Modern Standard Arabic and dialectal varieties, compounded by the limited availability of high-quality annotated datasets for children’s literature (Habash, 2010; Abdelghafur et al., 2025). While recent work has shown that models such as GPT-4 and Jais can generate fluent, EGRA-aligned stories, inconsistencies in narrative structure, deviations from target word counts, and repetitive outputs with minimal structural variation remain persistent issues (Abdelghafur et al., 2025; Rai et al., 2024; El-Shangiti et al., 2024).

2.3 Diversity vs. Constraint Trade-off

A central challenge in EGRA-based generation is balancing diversity with constraint satisfaction. Increasing diversity through stochastic decoding can lead to output degeneration and reduced control over pedagogical requirements (Holtzman et al., 2020), while overly constrained prompting often results in repetitive and structurally similar outputs (Rai et al., 2024).

Recent approaches attempt to address this trade-off. Shankarnarayanan et al. (2024) show that inspiration-based prompting can improve narrative diversity, while Madaan et al. (2023) demonstrate that iterative self-refinement can enhance output quality without model fine-tuning. Despite these advances, achieving consistent diversity while maintaining strict adherence to EGRA constraints remains unresolved, particularly in Arabic.

3 Related Works

3.1 Output-Level Diversity: Stochastic Decoding Strategies

A common approach to diversity in LLM generation is stochastic decoding, which modifies the token probability distribution at each step. Temperature scaling (Ackley et al., 1985) flattens or sharpens this distribution, while top- k sampling (Fan et al., 2018) restricts candidates to the k most probable tokens, and nucleus (top- p) sampling (Holtzman et al., 2020) selects the smallest token set whose cumulative probability exceeds a threshold. These methods trade output quality for diversity as their parameters increase, yet they share a fundamental limitation: all intervention occurs at the final vocabulary logit, leaving the model’s

internal computations, embeddings, attention patterns, and intermediate representations entirely unchanged. Locally Typical Sampling (Meister et al., 2023) approaches diversity differently by selecting tokens whose information content stays close to the conditional entropy of the model, reducing degenerate repetition while preserving fluency. A complementary line of work intervenes on the output vocabulary itself: Dineen et al. (2026) randomly mask logits during self-play training to stop the model from collapsing onto a narrow set of token sequences. Their mask is applied during RL training, while ours is applied at inference, making the two approaches complementary. We use temperature, top- p , top- k primary decoding baselines. Our internal noise methods are independent to all of them and can be combined with any output-level sampler.

3.2 Internal Noise Injection in Transformers

A smaller but closely related body of work explores injecting perturbations into the internal representations of transformer models rather than into the output distribution. Kang et al. (2024) is the most directly comparable prior work, they apply input-dependent Gaussian noise to the hidden states of early LLM layers during *fine-tuning* to produce diverse paraphrases of training examples for knowledge injection. Kang et al. (2024) demonstrate that latent-level perturbation yields more semantically diverse augmentations than data-level paraphrasing alone. This finding motivates our hypothesis that similar perturbations at *inference time* can promote creative diversity in generation. Unlike Kang et al. (2024)’s method of applying noise during training to improve knowledge retention, our methods apply noise at inference time to diversify outputs without modifying model weights.

The residual stream framework of Elhage et al. (2021) motivates our choice of injection sites. Embeddings, attention logits, and block outputs each represent distinct points at which information is written to the model’s shared communication channel, producing qualitatively different perturbation effects.

3.3 Entropy-Aware Generation

Several works use the model’s own uncertainty as a signal for adaptive decoding. EDT (Zhang et al., 2024) raises temperature when the output vocabulary entropy is low, encouraging exploration precisely when the model is overconfident. Our atten-

tion entropy-based method shares this motivation but monitors entropy over attention distributions rather than output logits, intervening at the attention level, before the model’s repetitive tendencies manifest in the output distribution.

3.4 Constrained Generation and Arabic Educational LLMs

Constrained decoding methods (Willard and Louf, 2023) enforce formal output constraints by masking invalid tokens at each step, but do so at the cost of forcing the model toward low-probability and often repetitive continuations. This phenomenon is seen clearly in Early-Grade Reading Assessment (EGRA) story generation (RTI International, 2016; Zuolkernan and Shapsough, 2024), where tight vocabulary and structural constraints narrow the valid token set and exacerbate the repetitive nature of outputs. Noise injection has been shown to systematically degrade a model’s ability to follow constraints and guidelines (Shahani et al., 2025), motivating careful calibration. Our noise decay strategy addresses this directly by reducing perturbation magnitude as generation progresses, allowing the model to explore creative directions early in the story while gradually restoring constraint following behaviour toward the conclusion. We evaluate story generation in this setting using a suite of small Arabic-centric and multilingual instruct models (Sengupta et al., 2023; Bari et al., 2024; Huang et al., 2024; Fanar Team et al., 2025), measuring diversity via Self-BLEU (Zhu et al., 2018) and semantic dispersion.

4 Noise Steering methods

4.1 Embedding Noise

Embedding noise injects Gaussian noise directly into the output of the token embedding lookup table during the decoding phase. This is the earliest possible injection site: the perturbed vector is the model’s sole representation of the newly generated token before any transformer computation has occurred. All subsequent hidden-layer inputs are equivalent to the previous block’s residual output and therefore have no independent embedding-level analogue.

Formally, let $\mathbf{e}_t \in \mathbb{R}^D$ denote the embedding of the token sampled at decode step t . We perturb it as:

$$\tilde{\mathbf{e}}_t = \mathbf{e}_t + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma(t)^2 \mathbf{I}) \quad (1)$$

where $\sigma(t) = \sigma_{\text{emb}} \cdot \delta(t)$ is the decayed standard deviation from Section 4.5. Noise is applied exclusively to decode steps, the prefill pass over the prompt is left unperturbed, ensuring that the model’s understanding of the instruction and constraints is not corrupted. Because the perturbation enters at the base of the residual stream, it is amplified and transformed by every subsequent attention and MLP layer, producing broad, high-level variation in the story’s trajectory.

4.2 Attention-logit Noise

Attention-logit noise injects Gaussian noise into the output of the self-attention sub-layer at selected transformer blocks, specifically after the output projection W_O but *before* the subsequent MLP sub-layer and before the residual addition. Within each targeted block the computation is:

$$\mathbf{x} = \text{LayerNorm}(\mathbf{h}^{(l-1)}) \quad (2)$$

$$\mathbf{a} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) VW_O + \varepsilon \quad (3)$$

$$\mathbf{h}^{(l)} = \mathbf{h}^{(l-1)} + \mathbf{a} + \text{MLP}\left(\mathbf{a} + \mathbf{h}^{(l-1)}\right) \quad (4)$$

$$\varepsilon \sim \mathcal{N}(0, \sigma(t)^2 \mathbf{I})$$

where $\mathbf{h}^{(l)}$ is the residual stream at layer l and $\sigma(t) = \sigma_{\text{attn}} \cdot \delta(t)$ follows the cosine decay schedule of Section 4.5. The KV cache is left intact: the last token position of the attention output tensor (shape $(B, 1, D)$ during decoding) is perturbed only, leaving all cached key and value states unchanged.

By targeting the attention branch’s contribution to the residual stream in isolation, before the MLP has processed the attended context, this method perturbs the information routing step of each selected block without corrupting the feed-forward transformation. This gives it a qualitatively different character from residual stream noise, which acts on the full block output after both sub-layers have run.

4.3 Residual Stream Noise

Residual stream noise injects Gaussian noise into the transformer’s residual stream after each selected block has completed its full computation of both the self-attention and MLP sub-layers. The noise at layer l and decode step t is:

$$\tilde{h}_t^{(l)} = h_t^{(l)} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma(t)^2)$$

where $h_t^{(l)}$ is the block output at the last token position and $\sigma(t) = \sigma_{\text{res}} \cdot \delta(t)$ is the decayed standard deviation from Section 4.5. Noise is applied only during the decoding phase when new tokens are being generated, leaving the prompt representation unperturbed. Injecting at the full block output means the noise enters the residual stream at the point where the block’s complete contribution has been written, and propagates forward through all subsequent layers.

4.4 Attention Entropy Noise Injection

Attention Entropy Noise Injection (AENI) conditions noise magnitude on the peakedness of the model’s own attention distribution at each decode step. The motivation, shared with EDT (Zhang et al., 2024), is that a model is most likely to produce repetitive or formulaic output precisely when it is most confident: attention heads that concentrate their weight on a small set of tokens signal over-reliance on a narrow context window, which in constrained story generation often corresponds to copying earlier story elements. AENI responds by injecting more noise when attention is peaked and less when it is diffuse, intervening adaptively rather than applying a fixed perturbation schedule.

At each targeted layer l and decode step t , we read the post-softmax attention weight tensor $\mathbf{W}^{(l)} \in \mathbb{R}^{B \times H \times T_k}$, where H is the number of heads and T_k is the current key-sequence length. We extract the last query row (the current token’s attention pattern) $\mathbf{w} = \mathbf{W}_{:, :, -1, :}^{(l)}$ and compute a peakedness score as the mean per-head maximum attention weight:

$$\phi(\mathbf{w}) = \text{mean}_h \max_j w_{h,j} \quad (5)$$

This score lies naturally in $[0, 1]$, is independent of context length, and requires no normalisation: a peaked head drives its maximum toward 1, while a uniform distribution drives it toward $1/T_k$. Gaussian noise is then added to the self-attention output after W_O but before the MLP and residual addition, with effective standard deviation:

$$\sigma_{\text{eff}}(t) = \sigma_{\text{aeni}} \cdot \phi(\mathbf{w}) \quad (6)$$

where σ_{aeni} is the per-model ceiling calibrated by RMS as in Section 5.1. As with attention-logit noise, only the last token position of the attention output (shape $(B, 1, D)$ during decoding) is perturbed and the KV cache is left intact.

4.5 Noise Decay

Applying constant noise throughout generation risks degrading constraint adherence, consistent with findings that activation-level noise systemically harms instruction and constraint following in LLMs (Shahani et al., 2025). We therefore apply a cosine decay schedule that reduces noise magnitude as generation progresses, allowing the model to explore creatively in the early tokens while recovering clean, constraint respecting decoding for the middle and conclusion of its story.

Let t denote the current decode step and T the decay horizon. The noise standard deviation at step t is scaled by:

$$\delta(t) = \frac{1}{2} \left(1 + \cos \left(\frac{\pi \cdot \min(t, T)}{T} \right) \right)$$

so that $\delta(0) = 1$ (full noise at the first generated token) and $\delta(T) = 0$ (no noise at and beyond the horizon), with the effective standard deviation given by $\sigma(t) = \sigma_{(t-1)} \cdot \delta(t)$.

5 Experimental Setup

5.1 Noise Magnitude and Model Calibration

A key challenge when applying noise across multiple models is that the scale of internal representations varies substantially between architectures and models (Dettmers et al., 2022; Sun et al., 2024). A fixed noise standard deviation that is undetectable for one model may be catastrophically large for another. To ensure that noise is applied at a consistent *relative* scale across all models, we calibrate the noise magnitude using the root mean square (RMS) of each model’s residual stream activations.

For each model we register forward hooks on nearly all transformer blocks (excluding the first and last layers) and collect block output tensors during *decode* steps only (i.e., when the sequence length is 1 and the KV cache is active), matching the phase at which noise is actually injected. The per-layer RMS is computed over the last generated token as:

$$\text{RMS}(l) = \sqrt{\frac{1}{C} \sum_C x^2} \quad (7)$$

where x is the block output tensor at layer l averaged over the batch and hidden dimensions, and C is the number of decode steps collected. We then aggregate these per-layer values into a single scalar

using the median across layers, which is robust to outlier layers at either extreme of the network.

The residual noise standard deviation for each model is then set as:

$$\sigma_{\text{res}} = \alpha \cdot \text{median}_l(\text{RMS}(l)) \quad (8)$$

where α is a fixed scaling coefficient set to 0.175 across all experiments. This formulation ensures that the injected noise is always proportional to the model’s own activation scale, making the effective perturbation strength comparable across models.

5.2 Generation

We compare each noise-injection method to the following baselines: a *noise-free* baseline ($T = 1.0$), a *High-temperature* ($T = 1.8$) baseline with $\text{top}k = 40$, and a *High-temperature* ($T = 1.8$) baseline with $\text{top}p = 0.9$. The truncation values follow prior work: Holtzman et al. (2020) report $k = 40$ in their main comparison and note that nucleus values “are usually in $[0.9, 1)$ ”. We pair these with an elevated $T = 1.8$, within the high-temperature regime explored by Nguyen et al. (2025). More aggressive settings (higher T , looser or no truncation) would likely produce still higher collapse rates, and absolute numbers would shift under per-model tuning. However, our principal claims concern the qualitative trade-off between output-level stochasticity and internal-representation perturbation, which we expect to be robust to such tuning.

Each condition generates 50 stories independently. Story i under every condition is generated with the same seed, which makes generation reproducible and aligns the underlying sampler RNG state across conditions where the sampling distribution coincides. However, since different conditions modify the sampling distribution itself (e.g. via temperature, $\text{top-}k$, $\text{top-}p$, or internal noise), the seed alone does not eliminate sampling variability between conditions. We therefore report all aggregate metrics over the full 50-story sample per condition and use the statistical tests in Appendix A to assess whether the observed effects are reliable rather than artefacts of sampling variance. All models are prompted with the same instruction to produce a story adhering to the EGRA constraints described in Section 5.3.3.

5.3 Evaluation

5.3.1 Creativity Scores

Vendi Score: The *Vendi Score* (Friedman and Dieng, 2023) measures the diversity of a set of items by quantifying how evenly distributed they are across predefined categories. For our stories, embeddings were obtained using the BAAI/bge-m3 model, with higher values indicating greater diversity.

Lexical Diversity: We compute Self-BLEU (Zhu et al., 2018), treating each story in turn as a candidate and the remaining 49 as references. The lexical diversity score is $1 - \text{Self-BLEU}$, so that higher values indicate less overlap across the generated set.

5.3.2 Readability and Linguistic Quality

We judge readability, grammatical correctness, linguistic quality, and appropriateness of reading level are evaluated using GPT5.3 Chat (OpenAI, 2026) as an LLM judge. Each story is scored independently on these dimensions, providing a complementary quality signal to the diversity metrics above.

5.3.3 Constraint Following

EGRA constraint adherence is evaluated through a combination of rule-based checks and LLM judgment. Constraints that are able to be verified exactly are checked programmatically. These constraints include: Story length not exceeding 60 words, Use of exactly one proper noun, Use of the present tense

Constraints that require semantic and structural understanding are evaluated using GPT5.3 CHAT (OpenAI, 2026) as a judge. These constraints include:

- Narrative structure includes intro, a middle dilemma, and an ending with resolution.
- Gender-balanced: includes both a boy and a girl.
- Avoids gender/religion/other stereotypes.
- Vocabulary suitable for children and the local context.

5.4 Models and Metrics

Models: We evaluate five small ($\leq 9\text{B}$ parameter) Arabic-centric instruct models. **Jais 8B** (Sengupta et al., 2023) is pretrained from scratch on a large Arabic and English corpus. **ALLaM 7B** (Bari et al., 2024) adapts LLaMA via Arabic tokeniser expansion and continued pretraining. **Fa-**

nar 9B (Fanar Team et al., 2025) emphasises morphology-aware Arabic tokenisation and dialectal coverage. **AceGPT** (Huang et al., 2024) is a bilingual (Arabic–English) model built on the LLaMA architecture with continued pretraining on large-scale Arabic data and instruction tuning for dialogue and generation tasks. **Phi-4-mini** is included as a compact multilingual baseline. Together these models span the main architectural and training strategies present in the current Arabic SLM landscape.

Quality (Qual. $\uparrow$). The quality score is an average of four LLM-judge dimensions assessed per story: overall readability, logical soundness, grammatical correctness, linguistic quality. Each dimension is scored independently out of 10 and the four scores are averaged into a single value per story, which is then averaged across the 50 stories in a condition.

Constraint Violations (Viol. $\downarrow$). We report the average number of constraints broken per story and the total constraints broken, where a lower score is better and zero indicates full compliance. This combines the programmatic and LLM-judge checks from Section 5.3.3: each of the seven constraints (three programmatic, four LLM-judged) contributes one violation point if failed, giving a maximum possible score of 7 per story.

6 Results

6.1 Total Modal Collapse

Before comparing diversity and quality metrics, we first examine instances of Total modal collapse (TMC): cases where generation degenerates into repetitive or incoherent token sequences, producing stories that are effectively unusable. Table 1 reports the collapse rate for each model and decoding condition. Any condition with a collapse rate at or above 40% is excluded from further evaluation, as the surviving outputs would constitute a

non-representative sample. Conditions with collapse rates greater than 20% (and less than 40%) are excluded from Figure 1 but are discussed further in Sec 6.2.

High-temperature baselines. The high temperature conditions (HiTemp-k and HiTemp-p) are the most fragile overall. AceGPT and Phi-4-mini both exceed the 40% threshold under both high-temperature settings, with collapse rates as high as 84% and 66% respectively. This confirms the well-known instability of aggressive output-level sampling: flattening the output distribution beyond a certain point causes the model to assign non-negligible probability to incoherent continuations, from which recovery is rare once the context has been corrupted (Nguyen et al., 2025).

Residual stream noise. residual stream noise (L-Res), records a 0% collapse rate across every model tested. Because this method injects noise after each transformer block has already completed its full computation, the perturbation enters the network at a point where the model’s internal representations are well-formed, and the signal propagates forward in a controlled manner. This makes residual noise a reliable location to inject noise without worrying about generation failures.

Attention noise and AENI. Raw attention-logit noise (Attn) shows high collapse rates, reaching 28% and 34% on ALLaM and Fanar respectively. However, our Attention Entropy Noise Injection method (AENI) eliminates collapse entirely across all five models. Rather than applying a fixed perturbation at every step, AENI scales the noise magnitude according to how peaked the model’s own attention distribution is at that moment. When the model attends broadly and diffusely, little or no noise is injected; perturbation is reserved for steps where attention has collapsed onto a narrow set of tokens, which is precisely when formulaic or repetitive output is most likely to follow. This adaptive

Table 1: Modal collapse rate (% of 50 generated stories) per model and decoding strategy. **Italic:** method excluded from further evaluation due to ($\geq 40\%$ collapse rate);

Model	Baseline	HiTemp-k	HiTemp-p	Embed	Attn	AENI	L-Res
ALLaM 7B	0%	0%	4%	22%	28%	0%	0%
AceGPT 8B	2%	84%	80%	0%	0%	0%	0%
Fanar 9B	0%	2%	0%	22%	34%	2%	0%
Jais 8B	0%	0%	0%	2%	4%	0%	0%
Phi-4-mini	0%	50%	66%	100%	44%	0%	0%

gating prevents the runaway instability that fixed attention noise produces.

Embedding noise. Embedding noise (Embed) causes complete collapse on Phi-4-mini (100%), and elevated rates on ALLaM (22%) and Fanar (22%). The Phi-4-mini model in particular is very sensitive to embedding noise, even at noise levels $\alpha < 0.175$ virtually every generation collapses. As a compact multilingual model, Phi-4-mini has had comparatively less Arabic training data, meaning its token embedding space for Arabic is more sparsely and less robustly organized. Because embedding noise acts at the very first step of the residual stream, before any transformer computation has occurred, any perturbation there propagates through every subsequent layer.

For a model whose Arabic representations are already fragile, this earliest injection site offers no opportunity for the model to compensate.

6.2 Diversity, Quality, and Constraint Adherence

Figure 1 plots each model-method condition in the space of Vendi Score (diversity, x-axis) against

LLM-judged quality (y-axis), with marker size encoding the average number of constraint violations per story. Beyond these core metrics, we also track estimated reading grade level per condition, as diversity methods that push stories toward richer vocabulary or more complex sentence structures may improve narrative variety and quality scores while silently violating the early-grade readability that EGRA passages require. Full numerical results are in Appendix Tables 6 and 4.

Residual Stream Noise. L-Res is the most consistently beneficial intervention. It achieves a 0% collapse rate for every model and improves Vendi Score in all five cases: +1.28 for ALLaM (7.14 $\rightarrow$ 8.42), +0.28 for AceGPT (9.95 $\rightarrow$ 10.23), +0.14 for Fanar (8.47 $\rightarrow$ 8.61), +2.38 for Jais (4.71 $\rightarrow$ 7.09), and +1.19 for Phi-4 (10.45 $\rightarrow$ 11.64). Constraint adherence is largely undisturbed, with violations changing by at most 0.78 per story across all models. Reading grade level is preserved throughout: ALLaM and Fanar remain at Grade 3, Jais at Grade 2, and AceGPT at Grade 4, all consistent with their respective baselines. These results indicate that

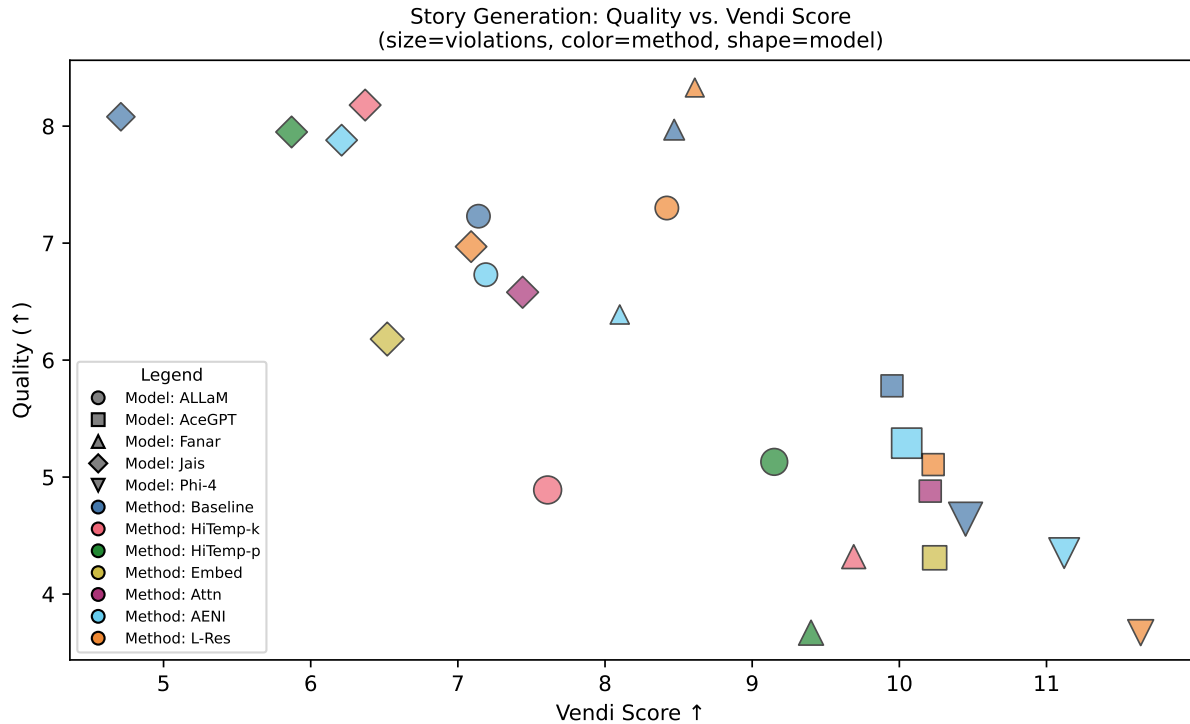


Figure 1: Story quality versus output diversity (Vendi Score) for all evaluated conditions with ≤ 10 total modal collapses. Each point represents one model–method combination: marker **shape** encodes the model and marker **colour** encodes the decoding method; marker **area** is proportional to the total number of EGRA constraint violations across the 50 generated stories (larger = more violations). Methods that cluster toward the lower-left or produce large markers trade quality or diversity for higher violation rates. Conditions with > 10 modal collapses are excluded to avoid diversity scores inflated by repetitive collapse outputs

perturbing the model’s internal representations allows it to diversify narrative choices, such as character, setting, and plot structure, without reaching for more complex language to do so. The quality impact depends on the baseline. For Fanar and ALLaM, which already produce strong stories (quality scores of 7.97 and 7.23 respectively), L-Res maintains or improves quality outright: Fanar reaches 8.33 while gaining diversity, placing it in a genuinely better region of the quality-diversity space. For weaker models the diversity gains come at a greater quality cost: AceGPT drops from 5.78 to 5.11, Jais from 8.08 to 6.97, and Phi-4 from 4.64 to 3.67. In Jais’s case the large Vendi Score gain suggests the model was simply not exploring the narrative space at baseline. In Phi-4’s case, even gentle perturbation strains its more limited Arabic generation capability, and grade level drifts to Grade 5, the one exception to L-Res’s otherwise consistent grade-level preservation.

Attention Entropy Noise Injection vs. Raw Attention Noise. Raw attention-logit noise (Attn) can achieve high diversity, with Vendi Scores of 9.51 for ALLaM and 7.44 for Jais, but at severe cost: collapse rates of 28% and 34% on ALLaM and Fanar respectively, constraint violations spiking to 5.36, and ALLaM quality collapsing to 2.55, the lowest value in the entire evaluation result in the surviving scores being a biased subset of an otherwise unreliable condition. AENI resolves this by conditioning noise magnitude on the model’s own attention entropy, injecting more perturbation when attention is sharply peaked and backing off when it is already diffuse. Collapse rates fall to 0% on ALLaM, AceGPT, and Jais, and just 2% on Fanar, with quality recovering substantially: ALLaM to 6.73, Fanar to 6.39, and Jais to 7.88. Crucially, AENI also preserves reading grade level across all Arabic-centric models and for Phi-4 actually improves it to Grade 3 from the baseline Grade 4, a stronger result than L-Res on that model. The cost of this stability is a moderate Vendi Score reduction relative to raw Attn, but this trade-off is clearly worthwhile for an educational generation system. On AceGPT, already a high-diversity model, the two methods produce near-identical Vendi Scores (10.21 vs. 10.05), confirming that AENI’s adaptive mechanism does not sacrifice diversity unnecessarily when it is not needed.

High-temperature Baselines High-temperature sampling is effective only for Jais, whose baseline

Vendi Score of 4.71 is the lowest in the evaluation. Both HiTemp-k and HiTemp-p raise diversity meaningfully (to 6.37 and 5.87) while quality stays high at 8.18 and 7.95, suggesting Jais’s cautious baseline was due to overfitting to the EGRA task format, producing structurally valid but narratively repetitive stories, and that output-level stochasticity was sufficient to break this pattern without destabilising the underlying generation quality. For every other model the picture is far worse: ALLaM’s quality falls to 4.89 and 5.13, Fanar’s to 4.32 and 3.67, and both AceGPT and Phi-4 exceed the collapse threshold entirely. The grade-level results are equally damaging: ALLaM’s Grade 3 baseline rises to Grade 6 and Grade 5 under the two conditions, and Fanar’s Grade 3 baseline drifts to Grade 5 and Grade 6 respectively. By flattening the output probability distribution, high-temperature sampling makes models more likely to reach for less common, more sophisticated vocabulary, precisely the kind of variation that is undesirable in an early-grade assessment context. It treats all forms of output variation as equivalent and has no mechanism for distinguishing a more creative plot from a more adult vocabulary, a distinction that is central to the EGRA task.

Appendix A contains significance tests that reaffirm these findings: L-RES and AENI are the only methods that consistently preserve quality and constraint adherence at a level statistically indistinguishable from the Baseline, while high-temperature sampling, embedding noise and raw attention noise injections are significantly worse on both metrics across the majority of models.

7 Conclusion

We investigated noise steering as a lightweight, fine-tuning-free approach to improving narrative diversity in Arabic EGRA-style story generation across five small language models. L-RES and AENI are the only methods that consistently improve diversity without the substantial degradation in quality, constraint adherence, or early-grade readability that the alternatives produce, the last of which proves a critical differentiator from high-temperature sampling, which inflates reading level and causes catastrophic collapse on several models. Residual stream noise is the most reliable all-around method and yields the greatest gains on models that already generate high-quality stories at baseline. AENI trades a small diversity mar-

gin for substantially better stability, making it the stronger choice when attention-level intervention is preferred. A consistent finding across all methods is that a model’s baseline generation quality is a strong predictor of how gracefully it absorbs noise: models with weaker Arabic representation degrade faster regardless of the intervention applied, suggesting that noise steering is best understood as a tool for unlocking latent diversity rather than compensating for limited language capability. Future work should explore layer-specific targeting strategies and extension to other low-resource languages and assessment frameworks.

Limitations

LLM-as-a-Judge Evaluation. Our quality and parts of our constraint adherence scores rely on GPT as an LLM judge, which introduces well-documented limitations. LLM judges are known to exhibit systematic biases including verbosity bias, where longer outputs receive inflated scores, and self-enhancement bias, where a model tends to favor outputs stylistically similar to its own generations (Chen et al., 2024). A particular concern in comparative evaluations is relative scoring bias, where a story may receive an inflated score simply by virtue of being better than the others it is presented alongside rather than on absolute merit. We took two steps to mitigate this: stories were submitted to the judge in fixed-size batches rather than all at once, and stories from different experimental conditions were shuffled together before evaluation, preventing the judge from implicitly comparing outputs within a single run. Quality dimensions such as narrative clarity and vocabulary suitability ideally require human annotators with relevant expertise. The programmatic constraint checks mitigate this concern for verifiable constraints such as word count and tense, but the LLM-judged dimensions: narrative structure, cultural neutrality, and reading appropriateness, should be interpreted with caution. Future work should incorporate human evaluation by Arabic literacy educators to validate the automated scores reported here.

Decoding-Hyperparameter Sensitivity. Our high-temperature baselines use single fixed values of $T = 1.8$, $k = 40$, and $p = 0.9$, chosen as widely used high-diversity defaults in prior work but not separately tuned per model. Since different models exhibit different baseline confidence pro-

files, the absolute diversity, quality, and collapse numbers for these baselines could shift under per-model hyperparameter sweeps. A similar caveat applies to the single value of $\alpha = 0.175$ used to calibrate noise magnitude across all noise-injection methods. We expect the qualitative ordering between output-level stochastic decoding and internal-representation perturbation to be robust, but a more comprehensive sensitivity study across decoding hyperparameters and per-model α values is left to future work.

Acknowledgments

This work was funded by the American University of Sharjah through FRG24-E-E87. The views and claims expressed in this work are those of the authors and do not necessarily reflect the official position of the University.

References

- Y. Abdelghafur, Y. Kaddoura, S. Shapsough, I. Zualkernan, and E. Kochmar. 2025. [Tales from the algorithm: Enhancing reading comprehension assessments with ai-generated arabic stories](#). In *Proceedings of the 10th International Congress on Information and Communication Technology (ICICT 2025)*, volume 1443 of *Lecture Notes in Networks and Systems*, pages 277–288.
- David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169.
- M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan AlRashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Al-rubaian, Ali Alammari, Zaki Alawami, and Abdulmohsen Al-Thubaity. 2024. ALLAM: Large language models for Arabic and English. *arXiv preprint arXiv:2407.15390*.
- Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. [Humans or LLMs as the judge? a study on judgement bias](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Process-*

- ing, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35.
- Jacob Dineen, Aswin RRV, Zhikun Xu, and Ben Zhou. 2026. [Vocabulary dropout for curriculum diversity in llm co-evolution](#). *Preprint*, arXiv:2604.03472.
- Margaret M. Dubeck and Amber Gove. 2015. The early grade reading assessment (EGRA): Its theoretical foundation, purpose, and limitations. *International Journal of Educational Development*, 40:315–322.
- Mahmoud El-Haj and Paul Rayson. 2016. [OS-MAN — a novel Arabic readability metric](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 250–255, Portorož, Slovenia. European Language Resources Association (ELRA).
- Ahmed O. El-Shangiti and 1 others. 2024. Arabic automatic story generation with large language models. In *Proceedings of the First Workshop on Arabic Natural Language Processing*.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. [A mathematical framework for transformer circuits](#). *Transformer Circuits Thread*.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, and Masoomali Fatehkia. 2025. Fanar: An Arabic-centric multimodal generative AI platform. *arXiv preprint arXiv:2501.13944*.
- Dan Friedman and Adji Bousso Dieng. 2023. [The vendi score: A diversity evaluation metric for machine learning](#). *Preprint*, arXiv:2210.02410.
- Nizar Y. Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2024. [AceGPT, localizing large language models in Arabic](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.
- Minki Kang, Sung Ju Hwang, Gibbeum Lee, and Jaewoong Cho. 2024. [Latent paraphrasing: Perturbation on layers improves knowledge injection in language models](#). In *Advances in Neural Information Processing Systems*, volume 37.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. [Locally typical sampling](#). *Transactions of the Association for Computational Linguistics*, 11:102–121.

- Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. [Turning up the heat: Min-p sampling for creative and coherent llm outputs](#). *Preprint*, arXiv:2407.01082.
- OpenAI. 2026. [GPT-5.3 Instant ChatGPT model](#). Accessed: 2026-03-06.
- Samarth Rai, Salsabeel Shapsough, and Imran Zualkernan. 2024. [Measuring fluency, coherency and logicity of GPT-4 generated EGRA comprehension stories](#). In *Proceedings of the 2024 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 201–203.
- RTI International. 2016. [Early grade reading assessment toolkit](#). Technical report, RTI International / USAID.
- Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Alham Fikri Aji, Zhengzhong Liu, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, and 3 others. 2023. [Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models](#). *arXiv preprint arXiv:2308.16149*.
- Prithviraj Singh Shahani and 1 others. 2025. [Noise injection systemically degrades large language model safety guardrails](#). *arXiv preprint arXiv:2505.13500*.
- Aadhith Shankarnarayanan, Taufiq Syed, Salsabeel Y. Shapsough, and Imran A. Zualkernan. 2024. [Once upon a GPT-4: Enhancing diversity in automated reading comprehension story generation with classic tales](#). In *Proceedings of the 2024 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 196–200.
- Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024. [Massive activations in large language models](#). *arXiv preprint arXiv:2402.17762*.
- Brandon T. Willard and Rémi Louf. 2023. [Efficient guided generation for large language models](#). *arXiv preprint arXiv:2307.09702*.
- Shimao Zhang, Yu Bao, and Shujian Huang. 2024. [EDT: Improving large language models’ generation by entropy-based dynamic temperature sampling](#). *arXiv preprint arXiv:2403.14541*.
- Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. [Texygen: A benchmarking platform for text generation models](#). In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1097–1100.
- Imran Zualkernan and Salsabeel Shapsough. 2024. [Towards using large language models to automatically generate reading comprehension assessments for early grade reading assessment](#). pages 3772–3782.

A Statistical Significance: L-Res and AENI vs. Alternatives

Test choice. We compare each method’s per-story quality and constraint-violation distributions to the Baseline using the Kruskal-Wallis omnibus test followed by Dunn’s pairwise post-hoc with Holm correction. This single non-parametric procedure is the appropriate choice for our data: Levene’s test rejects equality of variances ($p < 0.05$) in 7 of 10 (model, metric) groups, ruling out ANOVA + Tukey HSD; the metrics are also bounded and non-normal (quality $\in [0, 10]$ with TMC zero-mass; violations integer counts $\in [0, 7]$). Effect size is the matching non-parametric statistic, the rank-biserial correlation $r \in [-1, +1]$, with the convention that $r > 0$ indicates the method is *better than the Baseline* on that metric (quality higher, violations lower); $|r| < 0.1$ is negligible, ≥ 0.3 medium, ≥ 0.5 large, $|r| = 1$ is the maximum the metric supports.

Every alternative method significantly degrades performance whenever it can be tested. Tables 2 and 3 report rank-biserial r between each method and the Baseline for quality and violations respectively. Counting cells where the omnibus K-W is significant and the method survived TMC exclusion (the cells where Dunn’s post-hoc is licensed to run), the asymmetry between our proposed methods and the alternatives is severe: Embed produces a significant adverse deviation in **every single testable cell (6 of 6)**; Attn in 5 of 6; HiTemp- k and HiTemp- p in 3 of 5 each. By contrast, L-Res and AENI each produce a significant adverse deviation in only 2 of 7 testable cells. Where the alternatives register a significant effect, the median $|r|$ is 0.65 and *two separate cells reach* $|r| = 0.99$, *the maximum the metric supports*; where L-Res and AENI register one, $|r|$ never exceeds 0.48, and the median is 0.40. There is no model and no metric on which any alternative method is significantly closer to the Baseline than L-Res or AENI.

When alternatives degrade quality, they degrade it catastrophically. The visual pattern in Table 2 is unambiguous: every significant deviation produced by Attn, Embed, HiTemp- k , or HiTemp- p on quality is large or catastrophic ($|r| \geq 0.46$, with seven of fourteen at $|r| \geq 0.65$ and three at $|r| = 0.99$, where the metric saturates). The corresponding L-Res and AENI cells average

Table 2: **Quality.** Rank-biserial r between each method and the Baseline. Negative r indicates worse quality than Baseline. **Bold:** Dunn’s $p < .05$ (Holm-adjusted), with stars $*p < .05$, $**p < .01$, $***p < .001$. “—”: condition excluded by 40% TMC threshold. The Phi-4-mini omnibus K-W is non-significant ($H = 4.2$, $p = .12$); the column is shown for completeness but no Dunn’s test is licensed. Other omnibus K-W tests: ALLaM $H = 159.8$, AceGPT $H = 32.8$, Fanar $H = 158.9$, Jais $H = 62.8$ (all $p < .001$).

Method	ALLaM	AceGPT	Fanar	Jais	Phi-4
L-Res	−.02	−.30	+ .26	−.43**	−.22
AENI	−.39	−.28	−.48*	−.06	−.18
Attn	−.99***	−.46**	−.67***	−.48***	—
Embed	−.62***	−.61***	−.75***	−.57***	—
HiTemp- k	−.64***	—	−.81***	+ .01	—
HiTemp- p	−.65***	—	−.99***	−.06	—

Table 3: **Constraint Violations.** Rank-biserial r between each method and the Baseline. Negative r indicates more violations than Baseline. **Bold:** Dunn’s $p < .05$ (Holm-adjusted). “—”: condition excluded by 40% TMC threshold. AceGPT and Jais omnibus K-W are non-significant ($p = .19$ and $p = .08$); those columns show r values for completeness but no Dunn’s test is licensed. Other omnibus K-W tests: ALLaM $H = 80.9$, Fanar $H = 31.3$, Phi-4-mini $H = 13.3$ (all $p < .01$).

Method	ALLaM	AceGPT	Fanar	Jais	Phi-4
L-Res	+ .04	−.07	−.06	−.23	−.32**
AENI	+ .12	−.08	−.22	−.32	−.38**
Attn	−.69***	−.08	−.08	−.20	—
Embed	−.50***	−.25	−.33**	−.33	—
HiTemp- k	−.34*	—	−.30	−.21	—
HiTemp- p	−.25	—	−.51***	−.22	—

$|r| = 0.20$, well below the medium-effect threshold, and on several models L-Res sits within $|r| < 0.05$ of the Baseline distribution, indistinguishable from no intervention at all. The same pattern holds for violations (Table 3): the only large-magnitude deviation (−.69*** for Attn on ALLaM) belongs to an alternative method, and the only $|r| \geq 0.5$ result outside L-Res and AENI’s row is HiTemp- p on Fanar.

The L-Res and AENI exceptions. The two cells per method on which L-Res or AENI register a significant deviation each correspond to a known and bounded trade-off. L-Res on Jais quality ($r = −.43$, $p = .002$) is the model with the largest single diversity gain in the entire study (Vendi 4.71 $\rightarrow$ 7.09); a medium quality effect there is the cost of

the largest narrative-diversity improvement we report. AENI on Fanar quality ($r = -.48$, $p = .029$) sits well below the catastrophic $r = -.67$ that raw Attn produces on the same model, showing that AENI's adaptive gating recovers most of the quality that fixed attention noise destroys. Both methods also register a medium adverse effect on Phi-4-mini violations ($r = -.32$ and $-.38$); however, on this model L-Res and AENI are the *only two methods that survive TMC exclusion at all*. Embed, Attn, HiTemp- k , and HiTemp- p all collapse on $\geq 40\%$ of generations and are excluded entirely from this analysis. The choice on Phi-4-mini is therefore between a medium increase in violations under L-Res or AENI, and complete generation collapse under every alternative tested.

B Story Generation Prompt

System Prompt:

أنت مساعد مُفيد تُعدّ نصوص قراءة مُخصصة
للأطفال الصغار لتنمية مهاراتهم في فهم المقروء.

English Translation of System Prompt:

You are a helpful assistant who prepares reading texts specifically for young children to develop their reading comprehension skills.

User Prompt:

اكتب قصه.
* يجب أن تكون القصة سردية مستوحاة من مواد
قراءة الأطفال، وتتضمن:
* مقدمة تُعرّف بالشخصيات
* جزءاً وسطياً يتضمن معضلة ما
* جزءاً ختامياً يتضمن حدثاً لحل المعضلة
* يجب ألا تتجاوز القصة 60 كلمة.
* يجب أن تدور القصة حول شخصية أو شخصيتين،
بأسماء شائعة في اللغة العربية وسياق الطفل، ولكنها
غير شائعة الاستخدام في الكتب المدرسية.
* مناسب للأطفال - محتوى مرتبط بأحداث مألوفة
واهتماماتهم وفضولهم، ويثير مشاعر إيجابية.

* يحتوي على عناصر القصة القصيرة: شخصية، سياق،
بداية، عقبة أو مشكلة، وحل.

* متوازن بين الجنسين - يضم كلاً من الأولاد
والبنات.

* يتجنب الصور النمطية المتعلقة بالجنس أو الدين أو
غيرها.

* لا يوجد نص موجود مسبقاً ولا يذكر الأطفال
بخصوص أو أساطير يعرفونها.

* يستخدم زمن المضارع.

* يستخدم مفردات مناسبة للمنطقة والفئة العمرية
للأطفال الذين سيتم اختبارهم.

* يجب أن تكون الجملة الأولى سهلة للغاية. * يستخدم
بنية متنوعة ولكنها ليست أدبية أو معقدة.

* يستخدم اسماً واحداً (شائعاً) فقط.

* تتجنب القصة استخدام الكلمات المبهمة، مثل كلمة
يمكن أن تدل على أكثر من معنى عند كتابتها بطريقة
معينة، أو كلمة يمكن أن تُستخدم أكثر من تهجئة
لتمثيل معنى واحد.

* ليس قائمة من الجمل المترابطة بشكل ضعيف.

* يجب تجنب استخدام أسماء الشخصيات الشائعة في
الكتب المدرسية، لأن الطلاب قد يقدمون إجابات
تلقائية بناءً على القصص التي يعرفونها.

* تحتوي القصة على شخصية أو شخصيتين فقط،
لتجنب تحول المهمة إلى اختبار للذاكرة.

* يحتوي نص القصة على بعض المفردات المعقدة
وتراكيب الجمل.

اكتب القصة كنص سردي متصل دون عناوين أو
تسميات أو تقسيمات أو أي مؤشرات هيكلية.

English Translation of Constraints:

The story must be a narrative story generated from children's reading material which has a beginning section where the characters are introduced, a middle section containing some dilemma, and an ending section with an action resolving the dilemma.

The story should be 60 words long.

The story should revolve around one to two characters with names that are common to the Arabic language and context

of the child but not commonly used in school textbooks.

Appropriate for children - content related to familiar events, their interests, and their curiosity and evokes positive emotions

Has the elements of a short story: a character, context, beginning, obstacle or problem, and a resolution

Gender balanced – feature both boys and girls

Avoids gender, religious or other stereotypes

Does not already exist or remind children of stories or legends they already know

Uses the present tense

Uses vocabulary that is appropriate to the region and age of the children to be tested

The first sentence should be very easy

Uses varied structure but is not too literary/complicated Allows for a variety of comprehension questions (literal and inferential)

Only uses one (common) proper name

The story avoids the use of ambiguous words, such as a word that, spelled in one way, can represent more than one meaning or a word that can use more than one spelling to represent one meaning

It is not a list of loosely connected sentences.

Character names frequently used in the school textbook are to be avoided, as students may give automated responses based on the stories with which they are familiar

The story has only one to two characters, to avoid the task becoming about memory recall

The story text contains some complex vocabulary and sentence structures.

C LLM-as-a-judge prompt

The following prompt was given to GPT5.3 Chat, stories were given to the model in batches of 10,

with stories from different runs jumbled in each batch.

Listing 1: Judging prompt fed to GPT5.3 Chat

The following is a set of Arabic stories. I want you to rate each story out of 10 on the following metrics:

- Readability: How well does the story read, is it just a set of weakly connected sentences or does it flow well etc.
- Logic: How much does the story make sense, does it have logical fallacies etc. - Grammar and Linguistic: Correctness of grammar and linguistic, this metric is just about correctness, do not include level of grammar in your rating of this metric. Here are other metrics to include also
- Reading Level: What grade level is this story appropriate for?
- Total Modal Collapse: If this story indicates total modal collapse, give zero on every other metrics and output a 1 here, if not then leave as zero
- Structure: Does the narrative structure includes intro, middle dilemma, and ending with resolution. 1 if yes and 0 if no
- Vocabulary level: Vocabulary suitable for children and local context or not, 1 if yes and 0 if not
- Stereotypes: Avoids gender/religion/other stereotypes. 1 if yes and 0 if not
- Gender-balanced: includes both a boy and a girl. 1 if yes and 0 if not

Your response should be as a table format csv with the following column names:
Story number, Readability, Logic, GrammarandLinguistics, ReadingLevel, TotalModalCollapse, Structure, VocabularyLevel, Stereotypes, Gender-balanced”

D Readability and Grade Level Scores

Table 4 contains OsmanReadability Scores (El-Haj and Rayson, 2016) that were calculated to go along with the Grade level assessments made from our LLM-as-a-judge model.

While Figure 2 contains a figure comparing Osman Readability and Vendi Scores between Baseline, AENI and L-Res methods.

E Complete Diversity Scores

Table 5 contains the complete vendi and lexical diversity scores of methods that resulted in a Total modal collapse rate of less than 40%

F Quality Scores

Table 6 contains the complete quality scores for each run with the exception of Phi-4-mini Embedding Noise as it failed to produce any coherent output.

Table 4: Readability and constraint-violation metrics for all conditions with $< 40\%$ modal collapse. **Read.** $\uparrow$: mean OsmanReadability score (El-Haj and Rayson, 2016) $\pm$ std (higher = easier to read). **Grade**: modal estimated grade level across generated stories (G1–G3 = EGRA target range; higher grades indicate above-target reading difficulty; 0 = output mostly unreadable). **Viol.** $\downarrow$: mean EGRA constraint violations per story $\pm$ std. **Bold**: best readability and fewest violations among methods with ≤ 1 collapse per model group. † method has ≥ 1 collapse (count in Table 1).

Model	Method	Read. $\uparrow$	Grade	Viol. $\downarrow$
ALLaM 7B	Baseline	70.42 $\pm$ 9.40	G3	3.38 $\pm$ 0.67
	HiTemp- k	62.90 $\pm$ 20.30	G6	4.40 $\pm$ 1.55
	HiTemp- $p^\dagger$	76.57 $\pm$ 21.89	G5	4.10 $\pm$ 1.42
	Embed †	59.76 $\pm$ 117.64	0	4.92 $\pm$ 1.44
	Attn †	74.07 $\pm$ 32.86	0	5.36 $\pm$ 1.57
	AENI	39.78 $\pm$ 211.51	G3	3.22 $\pm$ 0.42
	L-Res	76.81 $\pm$ 10.86	G3	3.40 $\pm$ 0.86
	Baseline †	82.12 $\pm$ 6.59	G4	3.08 $\pm$ 1.05
AceGPT 8B	Embed	85.84 $\pm$ 7.29	G3	3.54 $\pm$ 1.13
	Attn	85.02 $\pm$ 6.39	G3	3.12 $\pm$ 0.87
	AENI	81.84 $\pm$ 9.43	G4	3.14 $\pm$ 0.88
	L-Res	85.48 $\pm$ 11.85	G4	3.10 $\pm$ 0.84
	Baseline	79.25 $\pm$ 6.33	G3	2.86 $\pm$ 0.57
Fonar 9B	HiTemp- $k^\dagger$	52.65 $\pm$ 16.99	G5	3.48 $\pm$ 1.11
	HiTemp- p	53.55 $\pm$ 17.34	G6	3.83 $\pm$ 1.14
	Embed †	42.94 $\pm$ 66.52	0	4.42 $\pm$ 1.94
	Attn †	82.95 $\pm$ 5.91	0	3.54 $\pm$ 1.49
	AENI †	80.56 $\pm$ 5.81	G3	3.32 $\pm$ 1.08
	L-Res	77.97 $\pm$ 5.26	G3	2.90 $\pm$ 0.46
	Baseline	85.63 $\pm$ 7.62	G2	2.74 $\pm$ 0.94
	HiTemp- k	84.21 $\pm$ 7.48	G2	3.18 $\pm$ 1.10
Jais 8B	HiTemp- p	84.85 $\pm$ 7.55	G2	3.18 $\pm$ 1.17
	Embed †	34.67 $\pm$ 177.19	G2	3.46 $\pm$ 1.07
	Attn †	78.63 $\pm$ 41.50	G2	3.18 $\pm$ 1.02
	AENI	85.45 $\pm$ 6.12	G2	3.32 $\pm$ 0.89
	L-Res	82.93 $\pm$ 10.13	G2	3.20 $\pm$ 1.04
	Baseline	82.02 $\pm$ 6.30	G4	3.16 $\pm$ 1.02
	AENI	80.59 $\pm$ 15.23	G3	3.96 $\pm$ 1.16
	L-Res	82.37 $\pm$ 7.50	G5	3.94 $\pm$ 1.42

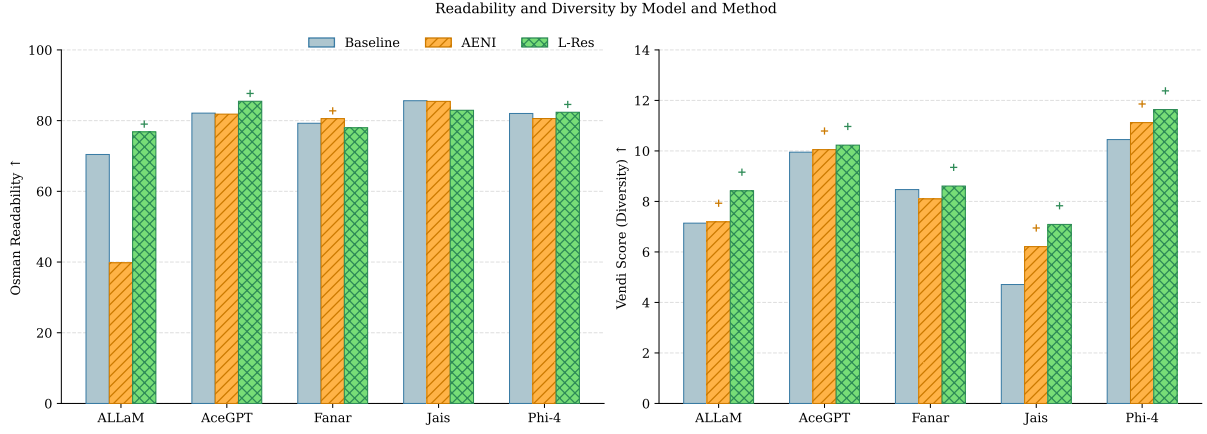


Figure 2: Osman readability score (left) and Vendi Score (right) for Baseline, AENI, and L-Res across all five models. A + marker above a bar indicates the method exceeds the Baseline for that model. Error bars are omitted for clarity.

Model	Method	VS	LD
ALLaM 7B	Baseline	7.14	0.748 ± 0.071
	HiTemp- k	7.61	0.949 ± 0.018
	HiTemp- p [†]	9.15	0.956 ± 0.013
	Embed [†]	6.92	0.838 ± 0.097
	Attn [†]	9.51	0.906 ± 0.036
	AENI	7.19	0.752 ± 0.119
	L-Res	8.42	0.833 ± 0.066

(a) ALLaM 7B

Model	Method	VS	LD
Fanar 9B	Baseline	8.47	0.939 ± 0.018
	HiTemp- k [†]	9.69	0.979 ± 0.005
	HiTemp- p	9.40	0.979 ± 0.005
	Embed [†]	8.92	0.976 ± 0.012
	Attn [†]	8.64	0.953 ± 0.013
	AENI [†]	8.10	0.950 ± 0.019
	L-Res	8.61	0.940 ± 0.024

(c) Fanar 9B

Model	Method	VS	LD
AceGPT 8B	Baseline [†]	9.95	0.942 ± 0.021
	Embed	10.24	0.943 ± 0.026
	Attn	10.21	0.940 ± 0.019
	AENI	10.05	0.944 ± 0.022
	L-Res	10.23	0.941 ± 0.022
	Phi-4-mini	10.45	0.949 ± 0.013
Phi-4-mini	AENI	11.12	0.954 ± 0.012
	L-Res	11.64	0.951 ± 0.017

(b) AceGPT 8B + Phi-4-mini

Model	Method	VS	LD
Jais 8B	Baseline	4.71	0.762 ± 0.139
	HiTemp- k	6.37	0.876 ± 0.067
	HiTemp- p	5.87	0.848 ± 0.076
	Embed [†]	6.52	0.916 ± 0.052
	Attn [†]	7.44	0.922 ± 0.031
	AENI	6.21	0.867 ± 0.056
	L-Res	7.09	0.928 ± 0.027

(d) Jais 8B

Table 5: Diversity metrics for all conditions with $< 40\%$ modal collapse. **VS**: Vendi Score. **LD**: Lexical Diversity ($1 - \text{Self-BLEU}$). **Bold**: best per model under ≤ 1 collapse. [†]: ≥ 1 collapse.

Model	Baseline	HiTemp-k	HiTemp-p	Embed	Attn
ALLaM	7.23 ± 0.81	4.89 ± 1.80	5.13 ± 1.97	4.09 ± 2.89	2.55 ± 1.24
AceGPT	5.78 ± 1.61	0.42 ± 1.03	0.60 ± 1.24	4.31 ± 1.50	4.88 ± 1.22
Fanar	7.97 ± 0.66	4.32 ± 2.00	3.67 ± 1.11	3.86 ± 2.74	4.77 ± 2.76
Jais	8.08 ± 0.92	8.18 ± 0.74	7.95 ± 0.75	6.18 ± 1.80	6.58 ± 1.85
Phi-4-mini	4.64 ± 1.18	2.59 ± 1.51	1.99 ± 0.70	–	2.41 ± 1.02

Model	AENI	L-Res
ALLaM	6.73 ± 1.05	7.30 ± 0.80
AceGPT	5.29 ± 1.38	5.11 ± 1.47
Fanar	6.39 ± 1.38	8.33 ± 0.71
Jais	7.88 ± 0.96	6.97 ± 1.36
Phi-4-mini	4.35 ± 1.29	3.67 ± 1.15

Table 6: Quality mean $\pm$ standard deviation for each model and method

The Effects of Structured LLM-Generated Feedback on Programming Assignment Performance

Tsvetomila Mihaylova¹, Evanfiya Logacheva¹, Arto Hellas¹, Jing Fan¹,
Francisco Castro², Bitu Akram³, Narges Norouzi⁴, Peter Brusilovsky⁵, Juho Leinonen¹

¹Aalto University, Department of Computer Science, ²New York University,

³North Carolina State University, ⁴UC Berkeley, ⁵University of Pittsburgh

Correspondence: tsvetomila.mihaylova@aalto.fi

Abstract

When programming students encounter errors in their code, compiler messages or static analysis output often provide limited guidance, particularly for novice programmers. Personalized feedback from instructors can be effective but does not scale well. Recent advances in large language models (LLMs) enable automated feedback generation at scale. This study examines whether LLM-generated feedback with different levels of guidance is associated with differences in students' problem-solving behavior. We analyze effects on time to solution and number of attempts, and examine whether these effects differ by programming experience. We design three feedback types and compare them to a baseline in which students receive only compiler error messages. Results from an online programming course show that LLM-generated feedback is associated with faster time to solution compared to the no-feedback baseline, with less guided feedback showing slightly stronger effects. Overall, the findings suggest that feedback structure plays an important role in how students progress toward correct solutions and motivate further work on adaptive feedback designs and longer-term learning outcomes.

1 Introduction

When students submit incorrect programming solutions, they typically receive feedback in the form of compiler errors or failed test cases (Keuning et al., 2018). Such feedback indicates that a solution is incorrect but often provides limited guidance on how to proceed, particularly for less experienced programmers (Becker, 2016). Human instructors can offer more targeted feedback, but this does not scale to large courses (Messer et al., 2024). Recent work explored the use of large language models (LLMs) to generate programming feedback automatically (Hellas et al., 2023; Mueller et al., 2025; Vassar et al., 2024; Gabbay and Cohen, 2024;

Koutchme et al., 2024, 2025). Prior studies report that LLMs can produce readable explanations and hints, but also note that generic LLM-generated feedback may be overly detailed, provide complete solutions, or contain inaccuracies (Xiao et al., 2024; Leinonen et al., 2023; Brown et al., 2025). Research in computing education further shows that novice and more experienced programmers differ in how they approach programming problems and respond to guidance (Chuang and Chang, 2024; Kalyuga, 2007).

This study examines how the structure of LLM-generated programming feedback is associated with student performance in an online programming course. Rather than focusing on fully personalized or adaptive feedback, we compare feedback formats that differ in the amount of guidance and information they provide, ranging from more detailed explanations to more constrained, diagnostic hints. We evaluate how these different feedback structures relate to students' time to solution and number of attempts, and examine whether their effects vary across students with different programming experience. In addition, we investigate whether the effectiveness of LLM-generated feedback differs across course content that varies in recency relative to the model training data, which may affect feedback quality.

We address the following research questions: **RQ1:** How does the different feedback complexity affect students' performance, in terms of (a) the expected number of attempts and (b) the time required to solve an assignment? **RQ2:** Do the effects of different hint types on attempts and solution time differ between students with different levels of expertise? **RQ3:** How does the recency of the instructional materials relate to students' performance, as measured by the expected number of attempts and time to solution when using generated hints?

The contributions of this work are as follows:

(1) We design programming assignment hints with different level of cognitive load, motivated by previous work of preferences for different proficiency levels. We show with experiments in an online programming course that LLM-generated hints lead to faster time to solution, with simple hints being more efficient. (2) We evaluate the performance of students of different programming proficiency when presented with the different hint types. (3) We show that new content, released after the LLM knowledge cut-off date, affects the quality of the hints negatively.

2 Background

2.1 Programming Feedback

Automated assessment systems usually provide feedback as test-case results, compiler diagnostics, or static-analysis warnings. This feedback is often difficult for novice programmers to interpret, especially when error messages rely on professional terminology or assume prior knowledge (Becker, 2016; Messer et al., 2024). Enhanced compiler error messages can improve readability, but prior work reports mixed effects on learning outcomes and notes challenges in scaling such approaches across courses and languages (Keuning et al., 2018). Human instructors can provide contextualized feedback adapted to a student’s understanding, but this does not scale to large courses. Large language models (LLMs) have therefore been explored for automatic feedback generation. Classroom studies report that LLM-generated hints are often perceived as more readable than compiler output and can support student progress across repeated submissions, while also revealing potential risks of over-reliance on generated feedback (Pankiewicz and Baker, 2023; Leinonen et al., 2022). Other evaluations report that LLM-generated feedback may include overly detailed guidance, complete solutions, or hallucinated issues, and that students often use such feedback primarily for immediate help-seeking rather than deeper self-regulatory strategies (Xiao et al., 2024; Viberg et al., 2025). Designing LLM-based feedback around pedagogical principles influences how students engage with generated hints, with planning-oriented support showing stronger associations with performance than other forms of assistance (Phung et al., 2025). Instructor-based evaluations also indicate that the quality and usefulness of LLM-generated programming feedback depend strongly on prompting strategies and

task difficulty, with substantial variation observed across zero-shot, one-shot, and few-shot configurations (Barros et al., 2025).

2.2 Novice vs. Advanced Programmers

Programming behavior differs across experience levels. Novice programmers spend more time debugging, make more syntax-related errors, and rely more on trial-and-error approaches. More experienced programmers make fewer errors and perform more targeted code modifications (Chuang and Chang, 2024; Shrestha et al., 2022). These differences are reported when proficiency is defined by prior experience rather than task performance (Chuang and Chang, 2024). The *expertise reversal effect* describes how instructional approaches that benefit learners with low prior knowledge may become less effective as expertise increases (Kalyuga, 2007, 2009). This effect has been discussed in programming education in relation to the usefulness of detailed guidance at different experience levels. More broadly, adapting the amount and form of instructional explanation to user expertise has been shown to improve learning outcomes in adaptive systems, suggesting that differences in feedback presentation across proficiency levels may be beneficial (Boyle and Encarnacion, 1994). Studies of automated program repair report that novice programmers often struggle to interpret repair suggestions, which may differ substantially from their original solutions, while more experienced learners report greater usefulness; students also tend to value error localization over full repair suggestions (Kurniawan et al., 2023). These studies also report that students often value information about the location of an error and that full repair suggestions can be difficult to apply (Kurniawan et al., 2023). Our work examines whether LLM-generated feedback that differs by programming proficiency and explanation depth is associated with differences in student performance.

3 Methodology

3.1 Context

We applied LLM-generated feedback on an online introductory Web Development course in the span of two months. When a student submits incorrect assignment solutions to the system, by default they see compiler error messages. We assigned at random for part of the users, one of the types of feedback messages. There was an indication that

the feedback was AI-generated, and the students had the option to vote for each message as Good or Poor.

3.2 Prompt Design

We used three types of prompts ([Appendix A](#)), adjusted for different proficiency levels. **General Feedback:** This prompt generates up to three issues in the code, each of which contains description, explanation and hints for next steps. **Single-Issue Feedback:** This prompt contains only one error, with its description, explanation and guidance for next steps towards the resolution. This is motivated by prior work which states that novice programmers could be overwhelmed by the feedback complexity ([Xiao et al., 2024](#)). **Diagnostic Feedback:** This prompt contains up to three issues with the code, but gives only descriptions, not hints for how to address the errors. This structure is motivated by the expertise reversal effect ([Kalyuga, 2009](#)), where more advanced programmers perform better with more limited guidance. We used GPT-4o for generating all feedback messages.

3.3 Data

The data used in this study is from an online introductory Web Development course. We only used submissions of students who allowed using their data for research. We excluded all exercise-user pairs for which at least one submission is longer than 2000 characters – to keep the prompt length shorter, no feedback is displayed for such submissions. We had three types of feedback, and one control group, (No Feedback), which did not see any LLM-generated messages. In addition to the LLM-generated feedback, the students also saw the errors logs from the failed automated tests.

We used the data from two parts of the course – *Part 2: Client-Side Development* and *Part 3: Server-Side Development*. Part 2 contains materials about more recent technologies, which were released after the used model cutoff date, and we refer to it as *New Content*. Therefore we also examined the results per course type, to explore whether the recency of the materials affected the generated hints. We refer to Part 3, which contains content that was used in the model training data, and we refer to it as *Old Content*.

We focused our investigation only on submissions for which the students were actively focused and worked on the task (i.e., engaged). Previous work defines that if a student does not do any input

on the programming task for more than 10 minutes, they are likely to be disengaged ([Shrestha et al., 2022](#)). Combining this insight together with the distribution of the number of exercises that were solved in particular time intervals, we analyzed assignments that were completed by a student for up to 30 minutes, as nearly 95% of all assignments were solved in this time interval.

Before starting the course, the students were asked the following question about their proficiency level: *Prior programming experience (1=less than 1 year, 2 = 1-2 years, 3 = 2-3 years, 4=3-5 years, 5=6-10 years, 6=more than 10 years)*, which they were not required to answer. Following previous research ([Chuang and Chang, 2024](#)), we split the students as *Novice Students* if their experience was below 3 years (answers 1, 2 or 3 from the above question), and as *Advanced Students* if they had over three years of experience (answers 4, 5 or 6 to the above question). When using this separation, there were 111 *Novice Students* and 38 *Advanced Students*.

After filtering out the submissions as described above, we obtained a dataset with 3153 assignment-user pairs, of which 1228 are *No Feedback*, 665 are *General Feedback*, 639 are *Single-Issue Feedback*, 621 are *Diagnostic Feedback*.

3.4 Approach

3.4.1 Modeling approach

To answer the three research questions, we conducted three experiments, each corresponding to one research question. All analyses were performed on *assignment–user pairs* that were eventually solved, considering submission sequences of up to 30 minutes, as motivated above. Each student sees the same feedback type for all their submissions to the same assignment. For each assignment attempted by the student, we assign the feedback type at random. This design choice was made, in order to not put any student in disadvantage of seeing only one feedback type, before we had any information about the quality of each type.

Because each student may submit multiple solutions to the same assignment and may receive different hints for the different assignments, observations cannot be assumed to be independent. We therefore employ mixed-effects models ([Lindstrom and Bates, 1998](#); [Brysbaert and Stevens, 2018](#)), which allow us to account for repeated measures at both the student and assignment levels. Across all

experiments, we include fixed effects for hint type and, where relevant, student level or course part, as well as random intercepts for students and assignments. In addition, we include random slopes for hint type at the student level, allowing the effect of hints to vary across students.

For time-based outcomes, we use linear mixed-effects models (LMMs) (Lindstrom and Bates, 1988; Brysbaert and Stevens, 2018) on log-transformed time. For count-based outcomes (number of attempts), we use negative binomial generalized linear mixed-effects models (NB-GLMMs) (Breslow and Clayton, 1993; Hilbe, 2011) with a log link to account for overdispersion. Linear mixed-effects models were fit using statsmodels version 0.14 (Seabold and Perktold, 2010). Negative binomial generalized linear mixed-effects models were fit using Bambi version 0.17 (Capretto et al., 2022).

3.4.2 Outcome variables

Each assignment–user pair contains the complete submission sequence up to the first correct solution. From this sequence, we derive two performance measures: **Time to Success**: the time elapsed (in seconds) between the first submission and the first correct submission. Due to right skew, this variable is log-transformed for modeling. **Attempts to Success**: the number of submissions required until the first correct solution. This variable is discrete and overdispersed and is therefore modeled using a negative binomial distribution.

3.4.3 General model structure

Let y_{ij} denote the outcome for student i on assignment j , and let η_{ij} denote the linear predictor:

$$\eta_{ij} = \beta_0 + \beta^T \mathbf{X}_{ij} + u_{0i} + u_{1i} \text{HintType}_{ij} + v_{0j}, \quad (1)$$

where $\mathbf{X}_{ij}$ contains the fixed effects relevant to the experiment, u_{0i} is a student-specific random intercept, u_{1i} is a student-specific random slope for hint type, and v_{0j} is an assignment-specific random intercept.

For Time to Success, we model:

$$\log(y_{ij}) = \eta_{ij} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2). \quad (2)$$

For Attempts to Success, we model:

$$y_{ij} \sim \text{NegBin}(\mu_{ij}, \alpha), \quad \log(\mu_{ij}) = \eta_{ij}, \quad (3)$$

where $\alpha > 0$ is the overdispersion parameter. The experiments below differ only in the fixed effects included in $\mathbf{X}_{ij}$.

3.4.4 Experiment 1: Overall effect of feedback on performance (RQ1)

To address RQ1, we include only hint type as a fixed effect:

$$\begin{aligned} \log(y_{ij}) = & \beta_0 + \beta_1 \text{HintType}_{ij} \\ & + u_{0i} + u_{1i} \text{HintType}_{ij} + v_{0j} + \varepsilon_{ij}. \end{aligned} \quad (4)$$

Here, β_0 represents the baseline performance under the no-hint condition, and β_1 captures the average effect of hint type. The same linear predictor is used for the NB-GLMM when modeling Attempts to Success.

3.4.5 Experiment 2: Effect of feedback by student expertise (RQ2)

To address RQ2, students are categorized as *novice* or *advanced*, and we include an interaction between hint type and student level:

$$\begin{aligned} \log(y_{ij}) = & \beta_0 + \beta_1 \text{HintType}_{ij} \\ & + \beta_2 \text{StudentLevel}_i \\ & + \beta_3 (\text{HintType}_{ij} \times \text{StudentLevel}_i) \\ & + u_{0i} + u_{1i} \text{HintType}_{ij} + v_{0j} + \varepsilon_{ij}. \end{aligned} \quad (5)$$

The interaction term β_3 captures whether the effect of hint type differs between novice and advanced students. An analogous NB-GLMM with the same linear predictor is used for Attempts to Success.

3.4.6 Experiment 3: Effect of feedback across course parts (RQ3)

To address RQ3, we include course part (used as a proxy for content recency) and its interaction with hint type:

$$\begin{aligned} \log(y_{ij}) = & \beta_0 + \beta_1 \text{HintType}_{ij} \\ & + \beta_2 \text{CoursePart}_{ij} \\ & + \beta_3 (\text{HintType}_{ij} \times \text{CoursePart}_{ij}) \\ & + u_{0i} + u_{1i} \text{HintType}_{ij} + v_{0j} + \varepsilon_{ij}. \end{aligned} \quad (6)$$

The interaction term tests whether the effect of hint type differs between earlier and later course parts. As before, Attempts to Success is modeled using an NB-GLMM with the same linear predictor and a negative binomial likelihood.

3.5 Ethics Statement

The work followed the ethical procedures at Aalto University.

4 Results

4.1 Experiment 1: Effect of hint types on the student performance on the assignment.

Time to Success Figure 1 summarizes the fixed effects of hint type on log-transformed time to success with the No Feedback condition as the reference level¹. All hint types were associated with a reduction in completion time relative to the No Feedback baseline. The largest average reduction was observed for Single-Issue Feedback ($\beta = -0.214$), followed by Diagnostic Feedback ($\beta = -0.194$) and General Feedback ($\beta = -0.122$). These effects are also reflected in the corresponding fixed-effect estimates shown in Figure 1. Beyond these average effects, the mixed-effects model accounted for substantial between-student and between-exercise variability. Random intercepts were included for both students and exercises, capturing baseline differences in performance. For time to success, the estimated variance of the student-level random intercept was $\sigma_{\text{student}}^2 = 0.103$, while the exercise-level random intercept variance was larger ($\sigma_{\text{exercise}}^2 = 1.025$), reflecting notable differences in task difficulty. In addition to random intercepts, the models allowed the effect of hint type to vary across students via student-specific random slopes. A wide spread of slopes was observed, which indicates considerable heterogeneity in how individual students responded to hints: while the average effect of hints was beneficial, some students benefited substantially more than others, and a smaller subset showed little or no improvement. This variability suggests that the reported fixed effects capture overall trends, but mask meaningful differences in individual responsiveness to hint support.

Attempts to Success The results for attempts, modeled using a NB-GLMM are shown in Table 1. The results indicate that all hint types credibly reduced the number of attempts to success relative to the No Feedback baseline, with the largest reduction observed for Diagnostic Feedback. The

¹Note: In all tables with results, we mark in bold the results that were statistically significant, i.e. the Confidence Interval (CI) for MixedLM or Highest Density Interval (HDI) for NB-GLMM excludes zero. Detailed numbers for time to success are given in the appendix.

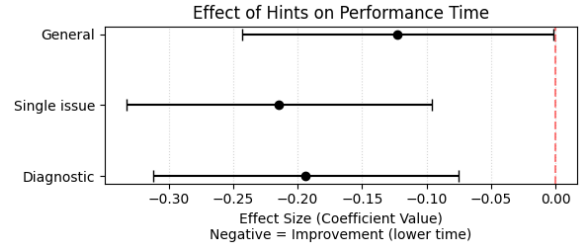


Figure 1: Effects of hint type on Time to Success, compared to the No Feedback baseline.

Table 1: Fixed effects of hint type on attempts to success (RQ1).

Predictor	β	94% HDI
Intercept	1.216	[1.130, 1.306]
General Feedback	-0.175	[-0.238, -0.113]
Single-Issue Feedback	-0.090	[-0.157, -0.030]
Diagnostic Feedback	-0.200	[-0.262, -0.134]

smallest for improvement is for Single-Issue Feedback, which could be explained with the fact that this hint type focuses on one issue only, and therefore might lead to a solution after more iterations. The standard deviation of the exercise-level random intercepts (≈ 0.26) reflects variability in baseline attempts across exercises, indicating differences in inherent task difficulty, while the standard deviation of the user-level random intercepts (≈ 0.17) indicates more moderate variability in baseline attempts across users. The standard deviations of the user-level random slopes for hint type were comparatively small ($\approx 0.05 - 0.07$), suggesting limited variation across users in how the effect of hints on attempts differs.

4.2 Experiment 2: Effect of the feedback for students with different expertise.

Time to Success To examine whether the effects of hint type differed between Novice Students and Advanced Students, we fit mixed-effects models including the interaction between hint type and student level for both Time to Success. Figure 2 shows the model-predicted outcomes. Single-Issue Feedback and Diagnostic Feedback are associated with improvement in the time for completion. In contrast, General Feedback does not show a clear effect. The main effect of student level is not statistically significant once hint type is accounted for. Although the interaction terms between hint type and student level are not statistically significant, the predicted trends suggest that the relative effec-

Table 2: Fixed effects of hint type, student level, and their interaction on attempts (RQ2).

Predictor	β	94% HDI
Intercept	1.228	[1.105, 1.334]
General Feedback	-0.196	[-0.325, -0.086]
Single-Issue Feedback	-0.220	[-0.339, -0.097]
Diagnostic Feedback	-0.207	[-0.319, -0.089]
Novice Students	-0.019	[-0.117, 0.078]
General Feedback $\times$ Novice Students	0.027	[-0.107, 0.168]
Single-Issue Feedback $\times$ Novice Students	0.174	[0.030, 0.316]
Diagnostic Feedback $\times$ Novice Students	0.007	[-0.132, 0.140]

tiveness of hint types might differ between Novice Students and Advanced Students.

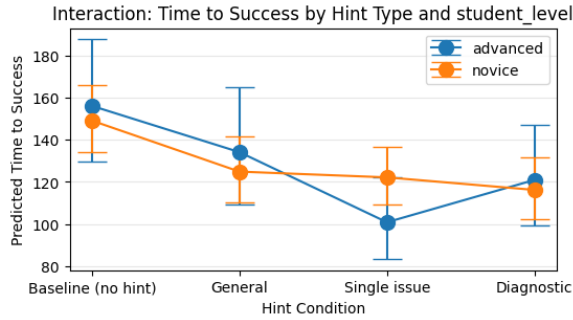


Figure 2: Interaction plot of hint type and student level, with predicted time to success from MixedLM model for assignments solved by up to 30 minutes.

Attempts to Success Table 2 shows the results of the negative binomial mixed-effects model examining the effects of hint type, student level, and their interaction on attempts to success, with No Feedback and Advanced Students as the reference categories. For Advanced Students (the reference group), all hint types were associated with a reduction in the number of attempts to success relative to No Feedback, with posterior mean effects ranging from $\beta = -0.20$ to -0.22 and 94% highest density intervals fully below zero. In contrast, baseline differences between Novice Students and Advanced Students in the No Feedback condition were small and uncertain. From these results it seems that Novice Students do not benefit from a particular hint type, and make slightly more submissions before submitting a successful solution when they see Single-Issue Feedback.

4.3 Experiment 3: Effect of the content recency on the quality of the LLM-generated feedback.

Time to Success To investigate whether the effectiveness of different hint types varies across course progression, we fit mixed-effects model including the interaction between hint type and course part for Time to Success. Model-predicted outcomes are shown in Figure 3. The reference level corresponds to assignments with No Feedback from New Content. All hint types for the Old Content perform better than New Content. All hint types show improvement over the No Feedback baseline for both course parts, with Single-Issue Feedback statistically significant for Old Content. The Diagnostic Feedback performs significantly better for New Content, likely due to the fact that the model could correctly identify errors in the submission, but if it tries to give a direction for fixing, and refers to older version of the material, this could lead to incorrect suggestions.

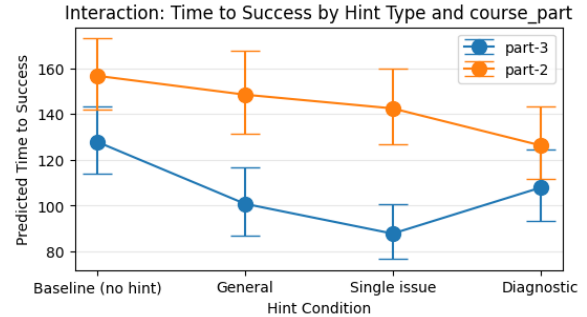


Figure 3: Interaction plot of hint type and course part, with predicted time to success from MixedLM model for assignments solved by up to 30 minutes.

Attempts to Success The results for attempts, modeled using a negative binomial mixed-effects model and shown in Table 3. Unlike the Time to Success, there is no observed significant difference in the Attempts to Success between the two parts when no LLM-generated feedback is shown. For the New Content, again the Diagnostic Feedback is associated with fewer attempts, but interestingly, the General Feedback, which also hints to solution, shows a reduced number of attempts. For the Old Content, the Single-Issue Feedback is associated with fewer Attempts to Success.

Table 3: Fixed effects of hint type, course part, and their interaction on attempts (RQ3).

Predictor	β	94% HDI
Intercept	1.177	[1.060, 1.292]
General Feedback	-0.129	[-0.211, -0.055]
Single-Issue Feedback	-0.034	[-0.120, 0.042]
Diagnostic Feedback	-0.217	[-0.300, -0.134]
Old Content	0.089	[-0.065, 0.257]
General Feedback $\times$ Old Content	-0.107	[-0.225, 0.007]
Single-Issue Feedback $\times$ Old Content	-0.131	[-0.248, -0.018]
Diagnostic Feedback $\times$ Old Content	0.037	[-0.088, 0.151]

4.4 Qualitative Analysis

In addition to the quantitative analysis of the student performance in terms of time and attempts to successful submission, we perform qualitative analysis, to identify common errors in the feedback generation. When the feedback was displayed to the students, it was accompanied by a message indicating LLM-generated content, and with option to mark the message as Good or Poor. Only very few of the feedback messages were evaluated. Most of the feedback that the users voted for was from Course Part 2 (New Content). There were only two Good and two Poor voted exercises for Course Part 3 (Old Content). The statistics of the given votes are shown in Table 5. The results in the table exclude one Good exercise that was not solved, and one Good exercise for which the correct solution was submitted 35 days after the first submission. Non-surprisingly, when Good feedback is given, the assignment is solved in a much fewer attempts and for much shorter time than when Poor feedback is shown. Students gave the positive vote in the first appearance of the feedback, and they waited for a few submissions before giving negative vote. Table 4 shows how many votes of each kind were given for the different types of feedback, but because of the small numbers, we cannot make any general conclusions from them.

Table 4: Number of Good and Poor votes by feedback type.

Hint Type	Good	Poor	Total
General Feedback	5	8	13
Single-Issue Feedback	7	12	19
Diagnostic Feedback	2	11	13
Total	14	31	45

We further perform thematic analysis of the Poor submission, in order to identify common errors. We first went through all the submissions voted as Poor and read them carefully in the context of the assignment description, correct submission and student submission. We also take into account the other submissions by the same user and the same assignment. Then, we coded each voted submission with a description of the type of error it contains. After this, we summarized the codes into common themes. We marked some feedback messages with more than one code, for example in cases where the LLM was referring to an old version, and having had access to the examples in the materials could have showed the expected correct implementation. Several patterns were observed: The suggested fixes were referring to an older version of the taught technology (n=12). The pointed issues could be considered meaningful, but examples in the materials were implementing the functionality in the same way as in the student submission, so there was no actual error that the LLM reported (n=12). The LLM found no issue with the submission, although the tests were failing (n=7). Cases where a wrong issue was proposed, and the real issue in the code was missed (n=4). The feedback was actually correct and the student applied it in their correct submission (n=3).

5 Discussion

Across the course, students receiving LLM-generated feedback reached correct solutions faster than those with compiler output alone, across all feedback types. Differences appear mainly in efficiency (time and attempts), not in final success, as solution rates are high in all conditions. Clear differences emerge between feedback structures. Single-Issue Feedback is associated with faster completion but not consistently fewer attempts, suggesting quicker progress without reducing submissions. Diagnostic Feedback leads to both fewer attempts and faster completion, indicating more efficient iteration, likely by helping students localize issues. General Feedback improves performance over no feedback but shows weaker effects than the more constrained formats. Overall, more detailed guidance does not necessarily improve short-term performance. Limiting feedback to fewer issues or problem identification alone was generally associated with more efficient progress. When accounting for student level,

Table 5: Characteristics of feedback voted as Good or Poor by students. Attempts and time are measured until the first correct submission. Vote position indicates the submission index at which the vote was given.

Vote	#Pairs	#Votes	Attempts to Solve		Time to Solve (s)		Vote Position	
			Mean $\pm$ SD	Median	Mean $\pm$ SD	Median	Mean $\pm$ SD	Median
Good	13	14	2.92 $\pm$ 1.04	3	279.10 $\pm$ 463.97	155.82	1.65 $\pm$ 1.14	1
Poor	22	31	13.95 $\pm$ 17.34	7.5	51288.33 $\pm$ 87726.14	997.92	19.45 $\pm$ 20.71	8

the models do not show strong interaction effects between feedback type and expertise. However, the predicted outcomes suggest that the relative ranking of feedback types differs between novice and advanced students, with feedback appearing more beneficial for Advanced Students than for Novice Students in our experiments. These results indicate potential differences in how students at different levels respond to feedback structure, but they do not provide strong evidence for level-specific optimal feedback. The effectiveness of feedback varied across the different course parts. The results suggest that feedback formats that rely on specific guidance may be more sensitive to limitations in the model coverage of newer technologies. The analyses show substantial variation across students and assignments. Individual students respond differently to the same feedback type, and assignment difficulty contributes strongly to performance differences. As a result, the reported effects represent average trends rather than uniform improvements. This indicates that a single feedback strategy is unlikely to be optimal for all students or tasks. The results motivate further work on adaptive feedback approaches that adjust feedback structure based on student behavior, task characteristics, or prior interaction patterns, rather than relying on fixed feedback formats. Finally, we analyzed LLM-generated feedback evaluated by students and qualitatively examined messages marked as Poor. Due to the small number of votes relative to total submissions, strong conclusions cannot be drawn. Nevertheless, observed failure cases indicate that hallucinations can reduce feedback quality. Providing relevant learning materials in the prompt may mitigate some issues by adding context, especially when the model’s knowledge is outdated. Including compiler error messages could also help guide the LLM toward key issues in the submission.

6 Conclusion and Future Work

This study examines how different structures of LLM-generated programming feedback affect student performance in an online course. All feedback types help students reach correct solutions faster than compiler output alone, with differences appearing in efficiency (time and attempts) rather than final success. Feedback structure influences progress: lower cognitive load (fewer issues or less guidance) leads to more efficient short-term problem solving. However, large individual differences between students outweigh effects of expertise level, suggesting that coarse measures like years of experience may be insufficient. Feedback effectiveness also varies with course content. Guidance-heavy feedback performs less consistently on newer material, likely due to outdated model knowledge, indicating that relying only on pretrained knowledge is insufficient for content-specific feedback.

Future work should examine long-term learning effects and consistent feedback strategies across courses. Given high variability between students, more fine-grained personalization should be explored, including adaptation based on task, history, and behavior. Student proficiency could be assessed more reliably (e.g., via diagnostics), and preferences for feedback detail studied independently of expertise. Integrating retrieval or up-to-date course materials into feedback generation may improve accuracy, especially for new technologies. To mitigate the errors caused by LLM hallucinations, automated fact-checking mechanisms could be deployed to verify the correctness of the LLM-generated feedback before it is displayed to the student. Finally, future studies should evaluate long-term learning outcomes beyond short-term performance.

7 Ethical Statement

Displaying of LLM-generated feedback carries all the risks of using output from generative AI. We have taken several precautions to mitigate these

risks. First, the current study was conducted after an ethical approval by our university. Second, we are displaying notifications to the students that the displayed messages are LLM-generated. Third, in order to not put in disadvantage students who only see a particular feedback type, we show different feedback types per student. For the analysis, we only process data from students who have explicitly agreed their data to be used for research. In this work, we explore performance on the task, which does not necessarily correspond to deep understanding. We are planning a longitudinal study to ensure that the displayed feedback actually improves the long-term learning outcomes for the students.

8 Limitations

This study has several limitations. First, students were exposed to different hint types across assignments, so assignment–user pairs are not independent. This design avoided disadvantaging students before knowing feedback effectiveness, but future work could assign a single hint type per student or systematically vary it to study personal preferences. Second, student proficiency is based on self-reported experience, which may not accurately reflect actual ability. Third, the study is conducted in a single course, limiting generalizability, and excludes submissions over 2000 characters, potentially omitting more complex cases. Finally, we focus on short-term performance rather than long-term learning, which should be addressed in future longitudinal studies.

9 Acknowledgments

This work was supported by Research Council of Finland grants #356114 and #367787.

References

- Juliana Barros, Laura O. Moraes, Fernanda Oliveira, and Carla A. D. M. Delgado. 2025. [Large language models generating feedback for students of introductory programming courses](#). In *26th International Conference on Artificial Intelligence in Education, AIED 2025, Part 2*, volume 15877 of *Lecture Notes in Computer Science*, pages 421–433. Springer.
- Brett A. Becker. 2016. [An effective approach to enhancing compiler error messages](#). In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE '16*, page 126–131, New York, NY, USA. Association for Computing Machinery.
- Craig Boyle and Antonio O. Encarnacion. 1994. Metadoc: an adaptive hypertext reading system. *User Modeling and User-Adapted Interaction*, 4(1):1–19.
- Norman E. Breslow and David G. Clayton. 1993. [Approximate inference in generalized linear mixed models](#). *Journal of the American Statistical Association*, 88(421):9–25.
- Neil CC Brown, Pierre Weill-Tessier, Juho Leinonen, Paul Denny, and Michael Kölling. 2025. Howzat? appealing to expert judgement for evaluating human and ai next-step hints for novice programmers. *ACM Transactions on Computing Education*, 25(3):1–43.
- Marc Brysbaert and Michaël Stevens. 2018. Power analysis and effect size in mixed effects models: A tutorial. *Journal of cognition*, 1(1):9.
- Tomás Capretto, Camen Piho, Ravin Kumar, Jacob Westfall, Tal Yarkoni, and Osvaldo A Martin. 2022. Bambi: A simple interface for fitting bayesian linear models in python. *Journal of Statistical Software*, 103:1–29.
- Yung-Ting Chuang and Hsin-Yu Chang. 2024. Analyzing novice and competent programmers’ problem-solving behaviors using an automated evaluation system. *Science of Computer Programming*, 237:103138.
- Hagit Gabbay and Anat Cohen. 2024. [Combining llm-generated and test-based feedback in a mooc for programming](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale, L@S '24*, page 177–187, New York, NY, USA. Association for Computing Machinery.
- Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutchme, 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*, pages 93–105.
- Joseph M Hilbe. 2011. *Negative binomial regression*. Cambridge University Press.
- Slava Kalyuga. 2007. Expertise reversal effect and its implications for learner-tailored instruction. *Educational psychology review*, 19(4):509–539.
- Slava Kalyuga. 2009. The expertise reversal effect. In *Managing cognitive load in adaptive multimedia learning*, pages 58–80. IGI Global Scientific Publishing.
- Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2018. [A systematic literature review of automated feedback generation for programming exercises](#). *ACM Trans. Comput. Educ.*, 19(1).
- Charles Koutchme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, Syed Ashraf, and Paul Denny. 2025. Evaluating language models for generating

- and judging programming feedback. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 624–630.
- 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 on Innovation and Technology in Computer Science Education V. 1*, pages 52–58. Association for Computing Machinery, New York, United States.
- Oka Kurniawan, Christopher M Poskitt, Ismam Al Hoque, Norman Tiong Seng Lee, Cyrille Jégourel, and Nachamma Sockalingam. 2023. How helpful do novice programmers find the feedback of an automated repair tool? In *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 1–6. IEEE.
- Juho Leinonen, Paul Denny, and Jacqueline Whalley. 2022. [A comparison of immediate and scheduled feedback in introductory programming projects](#). In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*, SIGCSE 2022, page 885–891, New York, NY, USA. Association for Computing Machinery.
- Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. 2023. Using large language models to enhance programming error messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 563–569.
- Mary J Lindstrom and Douglas M Bates. 1988. Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data. *Journal of the American Statistical Association*, 83(404):1014–1022.
- Marcus Messer, Neil C. C. Brown, Michael Kölling, and Miaojing Shi. 2024. [Automated grading and feedback tools for programming education: A systematic review](#). *ACM Trans. Comput. Educ.*, 24(1).
- Moritz Mueller, Corinna List, and Michael Kipp. 2025. [The power of context: An llm-based programming tutor with focused and proactive feedback](#). In *Proceedings of the 6th European Conference on Software Engineering Education*, ECSEE ’25, page 1–10, New York, NY, USA. Association for Computing Machinery.
- Maciej Pankiewicz and Ryan Baker. 2023. [Large language models \(gpt\) for automating feedback on programming assignments](#). In *31st International Conference on Computers in Education (ICCE’2023)*, pages 68–77.
- Tung Phung, Heeryung Choi, Mengyan Wu, Adish Singla, and Christopher Brooks. 2025. [Plan more, debug less: Applying metacognitive theory to ai-assisted programming education](#). In *26th International Conference on Artificial Intelligence in Education, AIED 2025, Part 1*, volume 15877 of *Lecture Notes in Computer Science*, pages 3–17. Springer.
- Skipper Seabold and Josef Perktold. 2010. [statsmodels: Econometric and statistical modeling with python](#). In *Proceedings of the 9th Python in Science Conference*, pages 92–96.
- Raj Shrestha, Juho Leinonen, Albina Zavgorodniaia, Arto Hellas, and John Edwards. 2022. Pausing while programming: Insights from keystroke analysis. in *2022 IEEE/ACM 44th international conference on software engineering: Software engineering education and training (icse-seet)*.
- Alexandra Vassar, Jake Renzella, Emily Ross, and Andrew Taylor. 2024. [Fine-tuning large language models for better programming error explanations](#). In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*, Koli Calling ’24, New York, NY, USA. Association for Computing Machinery.
- Olga Viberg, Jacqueline Wong, Yael Feldman-Maggor, Nora Dunder, and Carrie Demmans Epp. 2025. [Chatting with code: Exploring llms as learning partners in programming education](#). In *26th International Conference on Artificial Intelligence in Education, AIED 2025, Part 6*, volume 15882 of *Lecture Notes in Computer Science*, pages 453–461. Springer. Interesting analysis of students talk with ChatGPT from the prospect of SRL. Too many simple requests for the answer.
- Ruiwei Xiao, Xinying Hou, and John Stamper. 2024. Exploring how multiple levels of gpt-generated programming hints support or disappoint novices. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–10.

A Prompts and Examples of Feedback Messages

Figure 4 shows the prompts used for the three feedback types. For choosing these prompts, we experimented with several different variants, and ended up with this formulation, which outputs the most concise feedback messages. Examples for the three types of feedback messages are shown on Figure 5.

B Data Stats

The distribution of the number of exercises that were solved in particular time intervals is shown in Table 6.

The solution rates per hint type are shown in Table 8, and show that the hint does not affect whether the students solve the assignment.

General Feedback:	Single-Issue Feedback:	Diagnostic Feedback:
SYSTEM You are a programming feedback ↳ tutor. Identify the main ↳ problems in the student's ↳ code and provide short, ↳ direct fixes for each. Do not show or quote the sample ↳ solution. RULES - Refer to the student's ↳ identifiers. - Be concise and factual; no ↳ filler or repetition. - Include up to three important ↳ issues. - For each, give a short Problem ↳ / Fix / Why block. - No code snippets or solution ↳ hints. - Prioritize: 1) Spec/I/O issues, ↳ 2) Logic/runtime errors, 3) ↳ Edge/complexity issues, 4) ↳ Style (only if required). OUTPUT FORMAT Issue 1: Problem: ... Fix: ... Why: ... Issue 2: Problem: ... Fix: ... Why: ... (Up to 3 issues) CONTEXT Assignment: ↳ {assignment_description} Starter Code: {starter_code} Sample Solution: ↳ {sample_solution} Student Code: {student_code} TASK Give compact feedback listing up ↳ to three issues, each with ↳ Problem / Fix / Why.	SYSTEM You are a supportive programming ↳ tutor. Help a beginner fix ↳ one main problem in their ↳ code. Do not show or quote the sample ↳ solution. RULES - Focus on one clear issue. - Use the student's own variable ↳ and function names. - Keep it short, kind, and easy ↳ to follow. - No code or sample-solution ↳ hints. OUTPUT FORMAT Problem: (what's wrong) Fix: (one simple action) Why: (what it fixes or changes) CONTEXT Assignment: ↳ {assignment_description} Starter Code: {starter_code} Sample Solution: ↳ {sample_solution} Student Code: {student_code} TASK Give one short piece of ↳ beginner-friendly feedback ↳ following the format above.	SYSTEM You are a peer-level code ↳ reviewer. Identify the most ↳ relevant issues in the ↳ student's code precisely and ↳ concisely. Do not include or hint at the ↳ sample solution. RULES - Use the student's identifiers ↳ when referencing code. - Be factual and specific; no ↳ filler, no praise, no ↳ instructions. - Describe what is wrong, where, ↳ and what impact it has. - Do not restate the assignment ↳ or include any code. - Prioritize: 1) Spec/I/O issues, ↳ 2) Logic/runtime flaws, 3) ↳ Edge/complexity issues, 4) ↳ Style (only if required). OUTPUT FORMAT Issues: - Problem -> Location -> Impact - (Up to 3 bullets) CONTEXT Assignment: ↳ {assignment_description} Starter Code: {starter_code} Sample Solution: ↳ {sample_solution} Student Code: {student_code} TASK Give terse, peer-style feedback ↳ listing only the key issues ↳ as bullets.

Figure 4: The three prompt variants used in the study for the different types of feedback.

The descriptive statistics per feedback type are shown in Table 7. In our current setup, each user sees one unique hint type for each assignment that they attempt, but they can see different hint type for the different assignments. This table also shows descriptive statistics about the hint length and complexity, and the weighted Levenshtein distance between the change in two consecutive submissions. The three feedback types differ in their textual properties: General Feedback are substantially longer and more variable in length, Single-Issue Feedback are shorter and the easiest to read, and Diagnostic Feedback are sim-

ilarly short but more linguistically complex. The average code change between consecutive submissions is similar across hint-based conditions, with Single-Issue Feedback resulting in the smallest code changes, which aligns with their focus on addressing one issue at a time. The No Feedback baseline shows slightly larger and more variable code changes. Very similar solution rate was observed for all the feedback types, as well as the case where only compiler errors were shown, therefore the different feedback types did not affect whether the students solve the assignment.

General Feedback:

Issue 1:

Problem: The `

` element's → text is `"Text → transformations"` instead of → `"Reactive objects"`. Fix: Change the heading text to → `"Reactive objects"`. Why: The assignment requires an → exact main heading.

Issue 2:

Problem: The imported component
→ name
→ (`Transformations.svelte`)
→ does not match the
→ implemented file name
→ (`ReactiveCounter.svelte`).
Fix: Rename the component file
→ or adjust the import so they
→ match.
Why: Consistent naming is needed
→ to correctly render the
→ component.

Issue 3:

Problem: The paragraph for total
→ numbers incorrectly
→ references `numbers.length`
→ instead of
→ `statsState.numbers.length`.
Fix: Use
→ `statsState.numbers.length`
→ in the paragraph.
Why: It ensures the paragraph
→ displays the correct updated
→ value from the reactive
→ object.

Single-Issue Feedback:

Problem: The `if` condition

→ compares `request.method` to
→ `PEEK` without quotes.

Fix: Replace `PEEK` with

→ `"PEEK"` in the condition.

Why: This ensures that the code
→ correctly compares the
→ request method to the string
→ `"PEEK"`, which is necessary
→ for the condition to work as
→ intended.

Diagnostic Feedback:

Issues:

- Incorrect internal link text
→ -> `→ href="/contact">Contact</a>`
→ in line 16 -> The text
→ should be "Contact page",
→ affecting compliance with
→ assignment specifications.
- Unnecessary list wrapping
→ links -> `
` element
→ wrapping the `` tags in
→ lines 15-17 -> Deviates from
→ instructions, which require
→ the links to appear directly
→ below the paragraph without
→ being in a list.

Figure 5: Examples for each of the three variants of feedback used in the study.

C Additional Results

C.1 Experiment 1 Additional Results: Effect of hint types on the student performance on the assignment.

Table 9 shows the fixed effects of hint type on log-transformed time to success with the No Feedback condition as the reference level. Figure 6 shows the distribution of the random slope estimates per student for Time to Success.

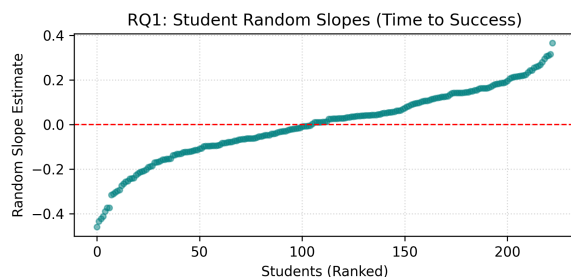


Figure 6: Student random slopes for Time to Success.

C.2 Experiment 2 Additional Results: Effect of the feedback for students with different expertise.

The fixed-effect estimates for log time to success with hint type $\times$ student level interaction are reported in Table 10.

C.3 Experiment 3 Additional Results: Effect of the content recency on the quality of the LLM-generated feedback.

The fixed-effect estimates for log time to success with hint type $\times$ course part interaction are shown in Table 11.

Table 6: Number of assignments that students completed in different time intervals.

Interval	No Feedback	General Feedback	Single-Issue Feedback	Diagnostic Feedback
≤ 10min	1068	587	585	563
10-20min	128	57	41	51
20-30min	32	21	13	7
30min-2h	47	20	15	19
2-24h	16	9	12	10
24-72h	9	3	2	4
> 72h	3	2	6	4

Table 7: Statistics of the data used in the experiments (assignments that were solved up to 30 minutes). Number of assignment-user pairs, submissions, users, and assignments, together with descriptive properties of the generated hints and subsequent code changes by hint type. Values are reported as mean $\pm$ standard deviation. Hint length is measured in words; hint readability is quantified using Flesch Reading Ease (higher values indicate easier readability); code diff is measured as weighted Levenshtein distance between consecutive submissions. There are 47 assignments in total, and each of the hint types was seen for all assignments.

Hint type	Pairs	Submissions	Users	Hint Length	Hint FRE	Code diff
No Feedback	1228	4677	206	–	–	0.074 ± 0.186
General Feedback	665	2208	161	88.78 ± 34.96	54.24 ± 11.12	0.071 ± 0.129
Single-Issue Feedback	639	2312	162	54.23 ± 10.95	57.43 ± 11.22	0.061 ± 0.129
Diagnostic Feedback	621	1966	162	56.30 ± 21.57	44.82 ± 14.42	0.072 ± 0.130
Total	3153	11163	223	–	–	–

Table 8: Number of assignment–user pairs per feedback type, showing total submissions, solved and unsolved cases, and the percentage of solved submissions.

Feedback type	All	Solved	Unsolved	% Solved
No Feedback	1328	1303	25	98.1
General Feedback	708	699	9	98.7
Single-Issue Feedback	684	674	10	98.5
Diagnostic Feedback	663	658	5	99.2
Total	3383	3334	49	98.6

Table 9: Fixed effects for log time to success (RQ1).
Reference level: No Feedback.

	β	95% CI	<i>p</i>
General Feedback	-0.122	[-0.243, -0.002]	0.046 *
Single-Issue Feedback	-0.214	[-0.333, -0.096]	0.000 ***
Diagnostic Feedback	-0.194	[-0.312, -0.075]	0.001 **

Table 10: Fixed effects for log time to success with hint type $\times$ student level interaction (RQ2).

	β	95% CI
General Feedback	-0.151	[-0.411, 0.108]
Single-Issue Feedback	-0.436	[-0.697, -0.176]
Diagnostic Feedback	-0.255	[-0.507, -0.002]
Novice Students	-0.046	[-0.259, 0.167]
General Feedback $\times$ Novice Students	-0.026	[-0.327, 0.276]
Single-Issue Feedback $\times$ Novice Students	0.237	[-0.064, 0.538]
Diagnostic Feedback $\times$ Novice Students	0.006	[-0.290, 0.302]

Table 11: Fixed effects for log time to success with hint type $\times$ course part interaction (RQ3).

	β	95% CI
General Feedback	-0.054	[-0.205, 0.097]
Single-Issue Feedback	-0.095	[-0.245, 0.055]
Diagnostic Feedback	-0.215	[-0.368, -0.062]
Old Content	-0.203	[-0.339, -0.067]
General Feedback $\times$ Old Content	-0.184	[-0.409, 0.042]
Single-Issue Feedback $\times$ Old Content	-0.281	[-0.503, -0.059]
Diagnostic Feedback $\times$ Old Content	0.045	[-0.181, 0.272]

Interpretable Difficulty-Aware Knowledge Tracing in Tutor-Student Dialogues

Shuyan Huang Alexander Scarlatos Jaewook Lee Andrew Lan

University of Massachusetts Amherst

{shuang, ajscarlatos, jaewooklee, andrewlan}@umass.edu

Abstract

Recent advances in large language models (LLMs) have led to the development of AI-powered tutoring systems that provide interactive support via dialogue. To enable these tutoring systems to provide personalized support, it is essential to assess student performance at each turn, motivating knowledge tracing (KT) in dialogue settings. However, existing dialogue-based KT approaches often ignore question difficulty and rely on opaque LLM latent representations, hindering accurate and interpretable prediction. In this work, we propose an interpretable difficulty-aware conversational KT framework that leverages LLMs to explicitly model student knowledge state and the difficulty of tutor-posed tasks at each dialogue turn. The framework incorporates the original question and the next tutor-posed task to estimate the student’s knowledge state and the difficulty of the upcoming turn. It further integrates Item Response Theory to map LLM outputs into student ability and question difficulty parameters, enabling interpretable prediction of student performance grounded in cognitive theories of learning. We evaluate the framework on two tutor-student dialogue datasets. Quantitative and qualitative results show that our framework outperforms existing KT baselines, meanwhile generating interpretable outputs consistent with cognitive theory¹.

1 Introduction

Tutoring, which provides tailored instruction and feedback to students in an interactive manner, has been shown to effectively improve learning outcomes (Nickow et al., 2020; Nye et al., 2014). With the advanced natural language understanding and generation capabilities of large language models (LLMs), researchers have developed an increasing number of LLM-based tutoring systems, such as

¹Our code and data are available at <https://github.com/umass-ml4ed/Difficulty-Aware-DialogKT>

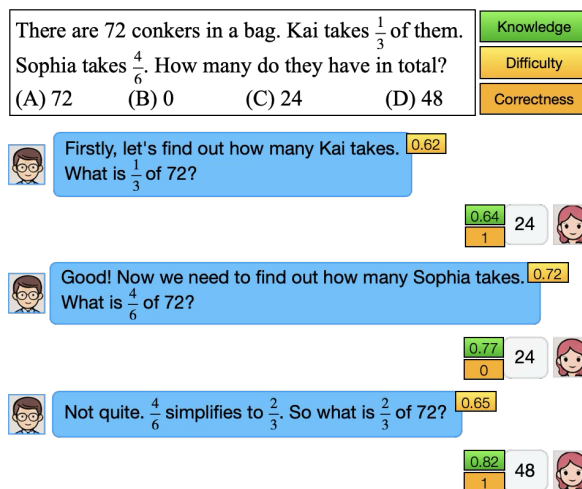


Figure 1: An example dialogue illustrating a student’s learning trajectory as the tutor adapts difficulty while scaffolding the math problem.

Khanmigo (Khan Academy, 2023) and LiveHint (Carnegie Learning, 2024). Recent work has further explored how to improve LLM-based tutors by training them to optimize student learning outcomes and by steering them toward diverse tutor personas that reflect different scaffolding, feedback, and affective-support styles (Scarlatos et al., 2025c; Lee et al., 2026). These systems can significantly reduce the workload of human tutors and provide scalable support for a large number of students.

To provide personalized pedagogical support, it is crucial to assess students’ mastery of concepts based on their responses throughout the dialogue. The process of modeling students’ knowledge states over time to predict their future performance is known as knowledge tracing (KT). Recently, with the development of AI-powered tutoring systems, KT has been extended to dialogue-based settings to predict whether students can correctly answer subsequent tutor-posed tasks (Scarlatos et al., 2025a; Wang et al., 2025).

However, existing KT approaches in dialogue

settings have two important limitations. First, prior studies do not explicitly model question difficulty. In practice, the correctness of student responses is not solely determined by their knowledge state but is also substantially influenced by question difficulty (Shen et al., 2022). Even students with high knowledge mastery may fail to answer very difficult questions. For example, as shown in Figure 1, although the student has a relatively high estimated knowledge state (e.g., 0.64), the student still answers the next tutor-posed task incorrectly when the question difficulty increases (e.g., from 0.62 to 0.72). Prior studies that model student responses solely based on students’ knowledge states may yield inaccurate predictions (Scarlatos et al., 2025a). Second, the reasoning process underlying model predictions of students’ future performance remains opaque. Most current approaches rely on LLMs with high-dimensional latent representations that lack interpretability. Consequently, it is difficult to derive psychologically meaningful explanations that align with cognitive theories of learning. This lack of transparency may undermine tutors’ trust in model predictions, hindering the adoption of LLMs as effective tutoring tools. As illustrated in Figure 1, even when the student answers a more difficult task incorrectly, the estimated knowledge state continues to increase, reflecting gradual learning dynamics consistent with cognitive theories (Newell and Rosenbloom, 1981; Anderson, 2000).

1.1 Contributions

To address the aforementioned challenges, we propose an interpretable difficulty-aware conversational KT framework built upon an LLM-based architecture that explicitly models both student ability and tutor-turn difficulty. Specifically, we prompt an LLM with the question and dialogue context to estimate student knowledge, and further prompt the model with the tutor turn to estimate the difficulty of the tutor-posed task. Furthermore, to bridge the gap between model predictions and cognitive theories of learning, we integrate principles from Item Response Theory (IRT). More specifically, we feed the estimated ability and difficulty into an IRT model to predict student correctness, enabling fine-tuning of the LLM using observed correctness labels. This design aligns model predictions with cognitive theory and improves interpretability by providing transparent reasoning based on the relationship between student ability and task difficulty. To evaluate the effectiveness of our approach, we

conduct comprehensive experiments on two tutor-student dialogue datasets. Both quantitative and qualitative results demonstrate the superiority of our framework in student ability and difficulty estimation throughout tutoring dialogues.

2 Problem Statement

Practically, a dialogue often starts when a student requests assistance from a tutor when they cannot successfully solve a math question/problem. Formally, we represent each dialogue $\mathcal{D}$ as an ordered sequence of alternating tutor turns t and student turns s , i.e., $\mathcal{D} = (s_0, t_1, s_1, \dots, t_N, s_N)$, where N denotes the total number of turn pairs in the dialogue and s_0 is included when the student initiates the dialogue. In cases where the same speaker produces several consecutive utterances, we merge them into a single tutor turn or student turn to maintain the alternating turn structure. Following Scarlatos et al. (2025a), each dialogue can be treated as a sequence of formative assessments, where each tutor turn t_j (for turn index j) acts as an assessment question, and the corresponding student turn s_j represents a correct or incorrect response. This perspective allows us to use tutor-student dialogues to model student learning over time. Each tutor turn with pedagogical intent (i.e., designed to assess student knowledge) is associated with one or more related knowledge components (KCs), $\mathcal{C}_j = \{c_j^k\}_{k=1}^{K_j}$, where K_j is the number of KCs associated with turn t_j . For student turns that follow these tutor turns, we define the correctness of a student turn s_j as $r_j \in \{0, 1\}$. In this way, we formulate the dialogue-based KT task as predicting the correctness $\hat{r}_{j+1}$ of the student’s next response with the given question q , historical dialogue $(t_1, s_1, \dots, t_j, s_j)$, and next tutor-posed task t_{j+1} , which can be formally defined as:

$$\hat{r}_{j+1} \sim f_{\Theta}(q, t_1, s_1, \dots, t_j, s_j, t_{j+1}), \quad (1)$$

where Θ denotes the learnable parameters of the framework, which are optimized to predict students’ future performance.

3 Approach

In this section, we detail our framework that consists of three modules: (1) a *knowledge estimator* that extracts latent knowledge representations from both the textual content of the question and the tutor-student dialogue history; (2) a *difficulty*

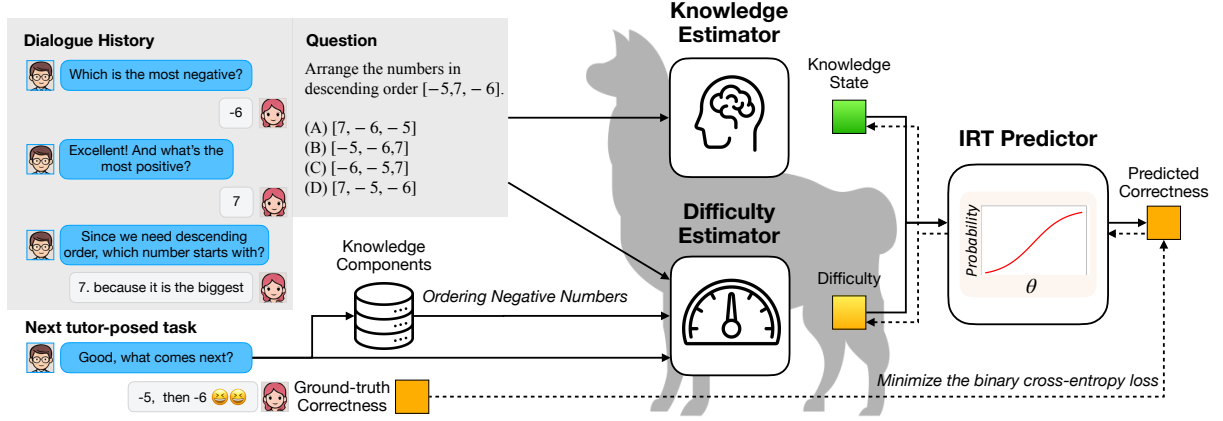


Figure 2: Overview of our framework with three modules: a knowledge estimator, a difficulty estimator, and an IRT-based predictor that jointly model student knowledge and task difficulty to predict correctness on the next task.

estimator that computes the difficulty of the tutor-posed task based on the question content, the dialogue history and content of the current turn; and (3) an *IRT-based predictor* that integrates the estimated knowledge state and difficulty representation to provide interpretable predictions of whether the student will correctly respond to the tutor-posed task. We illustrate our full framework in Figure 2, and show our prompts in Appendix A.

3.1 Knowledge Estimator

We now introduce the knowledge estimator module of our framework, which estimates student knowledge states for each turn in a dialogue. At each turn, the student’s subsequent performance depends on both the original question being solved and the dialogue history up to the current turn. The original question provides necessary context, while the dialogue history reveals the student’s evolving knowledge state, including observed errors or misconceptions. To effectively represent the textual information, we leverage the natural language understanding capabilities of LLMs to jointly encode the question text and dialogue history. To adapt general-purpose LLMs to the educational domain, we further fine-tune the model on dialogue-based tutoring data. We construct a prompt I_{j+1}^K representing turn $j + 1$, where we instruct the LLM to generate a one-word prediction for the student’s current ability level. Prior LLM-based KT work categorizes students’ knowledge states into discrete levels, e.g., good/fail (Li et al., 2025b). Inspired by this classification design, we use “GOOD” and “BAD” as positive and negative verbalizer tokens following prompt-based classification methods (Schick and Schütze, 2021). Rather than sampling text from the

LLM, we construct a continuous ability estimate by comparing their next-token logits. We first extract the LLM’s next-token output logits corresponding to these two verbalizer tokens:

$$z_{j+1}^{\text{GOOD}}, z_{j+1}^{\text{BAD}} = \text{LLM}(I_{j+1}^K).$$

We convert these logits into a scalar ability estimate by taking their difference, defining the student’s ability at the next turn as:

$$\theta_{j+1} = z_{j+1}^{\text{GOOD}} - z_{j+1}^{\text{BAD}},$$

where θ_{j+1} represents the student’s latent knowledge state at turn $j + 1$.

3.2 Difficulty Estimator

We now introduce the difficulty estimator module of our framework, which estimates the difficulty of each tutor-posed task in a dialogue. Difficulty estimation is recognized as a fundamental component in classical assessment settings, where accurately modeling item difficulty improves the reliability of student evaluation (Scarlatos et al., 2025b). In dialogue-based tutoring, each interaction is typically centered around an original question that the student is attempting to solve with tutor assistance. This original question is associated with an inherent difficulty determined by its textual content and cognitive complexity. As the dialogue progresses, the tutor can actively pose tasks with different difficulty in subsequent turns. Hence, to more accurately predict student performance at each turn, we estimate the difficulty of each tutor-posed task by jointly considering the original question text, dialogue context, and the next tutor-posed task. Similar to the knowledge estimator, we construct a prompt I_{j+1}^D

representing turn $j+1$, where we instruct the model to generate a one-word prediction for the difficulty level of the tutor-posed task. Again, we extract the LLM’s output logits, now corresponding to the “HARD” and “EASY” verbalizer tokens:

$$z_{j+1}^{\text{HARD}}, z_{j+1}^{\text{EASY}} = \text{LLM}(I_{j+1}^{\text{D}}).$$

We convert the logits into a scalar difficulty estimate, defining the difficulty of the tutor-posed task:

$$d_{j+1} = z_{j+1}^{\text{HARD}} - z_{j+1}^{\text{EASY}},$$

where d_{j+1} represents the latent difficulty of the tutor-posed task at turn $j+1$.

3.3 IRT-based Predictor

In prior LLM-based dialogue KT approaches, it is very challenging to explain the LLMs’ parameters and their decision-making processes when predicting student response correctness. Therefore, to align LLMs’ predictions with models in psychometric measurement, we introduce an IRT-based prediction layer that jointly predicts student response correctness while providing interpretable estimates of student knowledge and tutor turn difficulty. IRT, particularly the one-parameter logistic (1PL) model, also known as the Rasch model (Rasch, 1993), is widely used in educational assessment to model the probability of a correct response based on student ability and task difficulty. In the 1PL IRT model, student ability and task difficulty are represented as latent variables, and the probability of correctness increases as the positive difference between ability and difficulty grows. This formulation provides interpretability since both student knowledge and task difficulty are explicitly represented as meaningful cognitive quantities. Given the estimated ability θ_{j+1} and task difficulty d_{j+1} , the probability that a student will answer correctly at turn $j+1$ is defined as:

$$\hat{r}_{j+1} = \frac{1}{1 + \exp(-\alpha(\theta_{j+1} - d_{j+1}))},$$

where α is a learnable scalar that controls the sensitivity of the probability to differences between ability and difficulty. This parameter plays a role similar to the discrimination parameter in the two-parameter logistic (2PL) IRT model (Birnbbaum, 1968), enabling the model to adjust how strongly performance depends on the gap between ability and difficulty. Please note that we additionally experimented with modeling α as a learnable vector

to capture KC-specific discrimination, analogous to the 2PL model. However, this setting led to degraded performance in our experiments, likely due to overparameterization. Therefore, we adopt a scalar α in our final framework.

3.3.1 Model Optimization

To improve predictive accuracy, we fine-tune the LLM on correctness in observed tutor-student dialogues. We optimize the learnable parameters of the model by minimizing the binary cross-entropy loss between the ground-truth correctness of student responses r_{j+1} and the predicted probabilities $\hat{r}_{j+1}$ produced by the IRT-based prediction layer:

$$\mathcal{L} = -\sum (r_{j+1} \log \hat{r}_{j+1} + (1 - r_{j+1}) \log (1 - \hat{r}_{j+1})).$$

4 Experiments

4.1 Datasets

To evaluate the effectiveness of our framework, we conduct experiments on two datasets as follows:

Question-Anchored Tutoring Dialogues (QATD_{2k}) QATD_{2k} is the largest open-source real-world tutoring dialogue dataset (Zent et al., 2025). The dataset is collected from one-on-one conversations between tutors and students on the Eedi mathematics learning platform. Each dialogue begins when a student requests assistance in solving a math multiple-choice problem. After data preprocessing, the dataset contains 1,573 and 393 dialogues in the training set and test set, respectively. We use the LLM-generated KC and response correctness annotations collected in prior work (Scarlato et al., 2026).

MathDial MathDial is a tutoring dataset grounded in multi-step math reasoning problems between GPT-3.5-simulated students and crowd workers acting as tutors (Macina et al., 2025). Each dialogue begins with an incorrect student solution and the tutor guides the student to identify and correct their errors. The dataset contains 2,235 and 588 dialogues in the training set and test set, respectively. We use the LLM-generated KC and response correctness annotations collected in prior work (Scarlato et al., 2025a).

4.2 Baselines

We compare our framework against two types of representative KT baselines: LLM-based models and deep learning-based models.

4.2.1 LLM-based models

These models leverage LLMs to model student learning behavior by encoding question content and dialogue history to predict student responses.

- **LLMKT**: leverages textual question and conversation information in tutor–student dialogues using an LLM-based architecture, which is a previous state-of-the-art dialogue-based KT model (Scarlatos et al., 2025a).

4.2.2 Deep learning-based models

These models employ conventional deep neural networks to capture student knowledge states from sequences of their historical interactions, which consist of question tags, KC tags and binary responses, rather than LLMs.

- **DKT**: leverages an LSTM to sequentially model students’ learning processes and predict their future performance (Piech et al., 2015).
- **DKVMN**: leverages a static key memory matrix to encode relationships among KCs and a dynamic value memory matrix to predict students’ knowledge mastery levels (Zhang et al., 2017).
- **SAINT**: leverages a Transformer-based architecture to model students’ sequential learning interactions over time (Choi et al., 2020).
- **AKT**: leverages a monotonic attention mechanism to explicitly capture the temporal distance between current questions and students’ historical interactions (Ghosh et al., 2020).
- **simpleKT**: leverages a Rasch-inspired architecture to model question-specific variations and dot-product attention for modeling students’ sequential learning (Liu et al., 2023b).

4.3 Metrics and Experimental Setup

To quantitatively evaluate KT models in dialogue settings, we report Area Under the Curve (AUC) and Accuracy. We compute these metrics over all correctness labels except for the first label in each dialogue, since no prior observations are available for KT prediction at the initial step.

For our framework, we adopt Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the base model. We implement the model using the HuggingFace Transformers library (Wolf et al., 2020) and fine-tune it with LoRA (Hu et al.) on NVIDIA RTX L40 GPUs for efficient adaptation.

We perform a grid search to select hyperparameters for each model, choosing the checkpoint with the highest validation AUC to evaluate the test set. For LLM-based models, we search learning rates $\in \{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$ and LoRA ranks $\in \{16, 32\}$ and fixing the LoRA scaling factor $\alpha = 16$. The effective batch size is 64 via gradient accumulation, with gradient clipping at 1.0. We train the models for 5 epochs using BFloat16 mixed precision. For other KT baselines, we search learning rates $\in \{1 \times 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}\}$ and embedding sizes $\in \{8, 16, 32, 64, 128, 256, 512\}$. The batch size is 64 and the total epoch is 100, the remaining hyperparameters follow the default settings in pyKT (Liu et al., 2022). We train all models with AdamW optimizer with a weight decay of 1×10^{-2} .

5 Experimental Results

Model	QATD _{2k}		MathDial	
	Accuracy	AUC	Accuracy	AUC
Deep Learning-based Models				
DKT	60.19	54.73	59.75	63.14
DKVMN	58.24	51.79	56.42	60.64
SAINT	60.71	53.76	59.80	63.22
AKT	60.88	54.39	59.75	63.37
simpleKT	61.62	56.24	58.94	61.90
LLM-based Models				
LLMKT	<u>64.24</u>	<u>64.89</u>	<u>68.64</u>	<u>75.99</u>
Ours	64.29	65.25	68.82	76.59

Table 1: Student dialogue turn correctness prediction performance on QATD_{2k} and MathDial datasets. Best results are shown in **bold** and second-best results are underlined.

5.1 Quantitative Results

Table 1 shows the overall performance in terms of AUC and Accuracy for all models. We have the following observations: First, our proposed framework achieves the best overall performance and improves upon the previous state-of-the-art LLMKT model by explicitly modeling student ability, item difficulty, and combining them with IRT-based prediction. Specifically, our framework improves AUC from 64.89 to 65.25 (+0.36) on QATD_{2k} and from 75.99 to 76.59 (+0.60) on MathDial. These improvements suggest that explicitly

modeling question difficulty provides additional useful information for monitoring student learning progress. Although the performance gains are relatively modest, our framework introduces interpretable outputs, which we further analyze model interpretability in Section 5.3; Second, in general, LLM-based methods consistently outperform deep learning-based KT models in terms of both AUC and Accuracy, demonstrating the advantage of leveraging pretrained LLMs in dialogue-based KT settings. We attribute this improvement to the ability of pretrained LLMs to model rich textual content in dialogues, enabling us to better track student learning after fine-tuning on response correctness labels. Third, among deep learning-based KT methods, DKT achieves comparable performance with those complex architectures such as SAINT and AKT in both datasets. These results suggest that increasing architectural complexity does not necessarily yield significant performance gains. Instead, simpler architectures may generalize more effectively in dialogue-based KT settings, particularly under limited-data conditions (fewer than 2,500 dialogues), where complex models are more susceptible to overfitting.

5.2 Qualitative Analysis: Difficulty Analysis

To show that the learned difficulty parameters are indeed meaningful and interpretable, we measure the consistency between predicted KC difficulty and “ground-truth” KC difficulty. We use QATD_{2k} instead of MathDial for this analysis, since QATD_{2k} is collected from real-world educational environments and therefore better captures authentic tutor-student behaviors. Similar to (Lee et al., 2024), we follow classical test theory to define the ground-truth KC difficulty based on empirical correctness statistics as $1 - \frac{N_{\text{correct}}}{N_{\text{total}}}$, i.e., the portion of incorrect responses on this KC (Brown, 2013). To ensure a fair comparison, we normalize the predicted difficulty values using min-max normalization. As illustrated in Figure 3, the fitted regression line follows a similar increasing trend to the perfect agreement line, indicating that the predicted difficulty generally increases as the ground-truth difficulty increases. To quantitatively support this observed trend, we compute the Pearson correlation between predicted and ground-truth difficulty values across all KCs (Benesty et al., 2009). Results show a moderate positive correlation ($r = 0.368$, $p < 0.001$), suggesting that the learned difficulty parameters capture meaningful

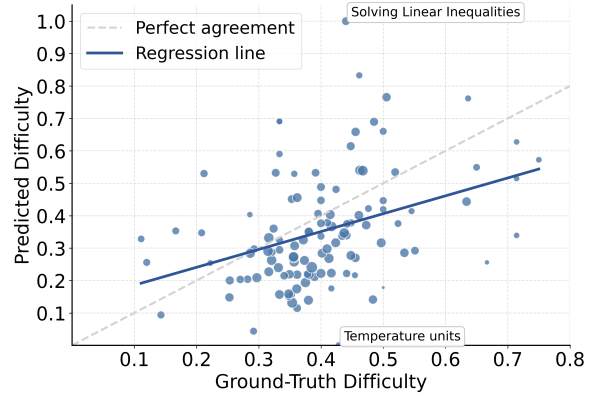


Figure 3: Comparison of predicted and ground-truth KC difficulty across KCs in the test set of the QATD_{2k} dataset. Each point represents one KC. Point size is proportional to the number of samples associated with each KC. The dashed gray line indicates perfect agreement, while the solid blue line shows the fitted linear regression trend.

empirical difficulty patterns in real-world tutoring data across KCs.

5.3 “Learning Curve” Analysis

A common approach to evaluate KT quality is to examine whether predicted learning dynamics align with cognitive theories such as the power law of practice (Snoddy, 1926). Following prior work (Shi et al., 2023; Scarlatos et al., 2025a), we visualize learning trajectories of each KC. We use QATD_{2k} instead of MathDial for this analysis, since QATD_{2k} is collected from real-world educational environments and therefore better captures authentic tutor-student behaviors. Specifically, for each KC, we group student interactions by the number of prior opportunities to practice that KC, where an “occurrence” refers to each time the KC appears in a tutor-posed task within a dialogue. For each occurrence index, we compute the average predicted mastery across all corresponding dialogue turns in the test set. The resulting curves therefore illustrate how the estimated mastery evolves as students encounter the same KC multiple times.

We examine plots of the top three most frequent KCs, as they contain sufficient samples and provide the most intuitive visualization of learning trends. As shown in Figure 4, LLMKT, which directly treats predicted correctness probability as knowledge mastery, exhibits noticeable oscillations across KC occurrences. The predicted probability decreases even after repeated practice, suggesting unstable representations of student mastery. In con-

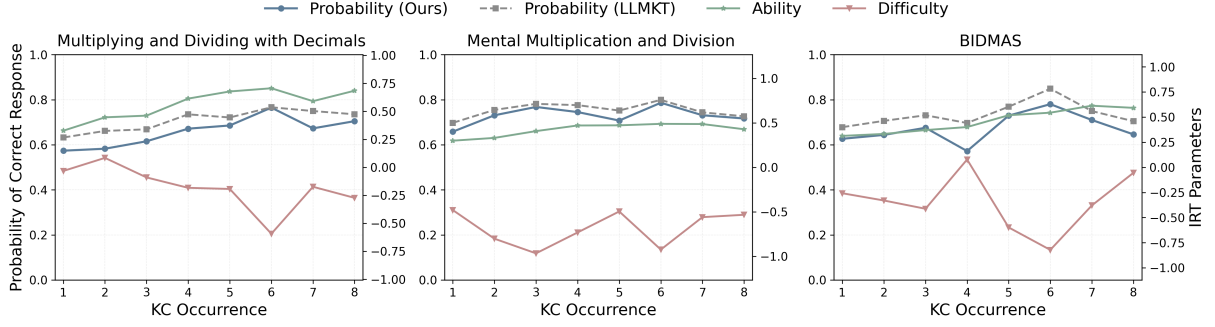


Figure 4: Learning trajectories for the three most frequent KCs in the QATD_{2k} dataset.

trast, our framework separates ability estimation from correctness prediction, resulting in smoother and more stable learning trajectories. The estimated student ability consistently increases with repeated KC occurrences across all three KCs, reflecting an increase in knowledge mastery that aligns with the cognitive theories of gradual learning (Ritter and Schooler, 2002).

Furthermore, fluctuations in predicted correctness probability from our framework are largely associated with changes in tutor-posed task difficulty. For example, temporary drops in predicted correctness probability often coincide with increases in estimated task difficulty, while the underlying ability continues to increase. Such fluctuations are expected in dialogue-based tutoring settings, where task difficulty may decrease when the tutor provides hints or scaffolding, and increase when the tutor switches to a different, more difficult part of the question or presses the student for a complete solution. Hence, our framework reflects realistic interactions between student learning progress and task difficulty. Similar trends observed across multiple KCs further demonstrate that our framework produces more stable and cognitively plausible learning dynamics than probability-based mastery estimation models such as LLMKT. We also provide learning trend visualizations at the dialogue level in Appendix B.

5.4 Qualitative Analysis: Case Study

We further conduct a qualitative case study to examine how the difficulty of tutor-posed tasks and the student’s knowledge state evolve in a dialogue in the QATD_{2k} dataset. Specifically, we analyze one representative dialogue from the test set. As shown in Table 2, the tutor dynamically adjusts the difficulty of the posed tasks based on the student’s responses. When the student fails to answer

correctly, the tutor decomposes the original question into simpler sub-tasks, leading to a decrease in the estimated task difficulty, from 0.671 to 0.360. In contrast, when the student answers correctly, the tutor proceeds with follow-up questions that maintain or slightly increase the difficulty level, moving the dialogue forward. Meanwhile, the student’s knowledge state demonstrates a generally increasing trend across the dialogue. Notably, even when incorrect responses occur, the estimated student ability continues to improve gradually with the tutor’s guidance, from 0.224 to 0.628 across the full dialogue. This observation highlights that learning progress may still occur despite temporary errors, as scaffolding and intermediate feedback from the tutor help consolidate the student’s KC understanding. Overall, this case study illustrates that our framework captures realistic tutoring dynamics, where task difficulty adapts to student performance while student ability evolves steadily through guided interaction.

6 Related Work

6.1 Student Modeling in Dialogues

Recently, with the emergence of LLM-based tutoring systems, several studies have explored KT in dialogue settings. Scarlatos et al. (2025a) leveraged the textual content in dialogues to estimate student knowledge. Wang et al. (2025) proposed a workflow that integrates KT with turn-by-turn verification to improve coding tutoring (Wang et al., 2025). However, these approaches overlook the modeling of question difficulty, which plays a critical role in KT. On the other hand, our approach introduces a difficulty-aware dialogue-based KT framework that explicitly aligns the student knowledge state and tutor-posed task difficulty, enabling fine-grained modeling of student state evolution. Beyond KT, recent studies have increasingly ex-

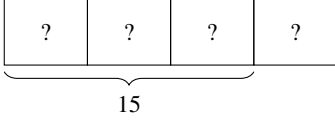
q	t_j	d_j	s_j	θ_j	r_j
Which question could the bar model represent?	what is the bar model split up into?	–	15	0.224	0
	and in terms of the whole bar model, how much does the 15 cover?	0.671	half.	0.475	0
	not quite! How many boxes is the bar split into?	0.360	4	0.531	1
A: Calculate $\frac{3}{4}$ of 15	okay good, and how many does the 15 span across?	0.447	All of the bar	0.605	0
B: $\frac{3}{4}$ of $\square = 15$	as in the bottom arrow. How many boxes is it under?	0.440	1	0.628	0
C: Calculate $\frac{1}{3}$ of 15					
D: $\frac{4}{3}$ of $\square = 15$	not quite, can you see that the 15 and the arrow go across 3 boxes?	0.405	–	–	–

Table 2: Case study of an adaptive tutoring dialogue illustrating how task difficulty (d_j) and student ability (θ_j) evolve over successive interactions in the QATD_{2k} dataset. The tutor dynamically scaffolds the problem by decomposing it into simpler sub-tasks following incorrect responses, while progressively maintaining or increasing difficulty after correct answers, leading to a gradual improvement in the student’s knowledge state.

plored LLM-based student simulation in educational dialogue settings. Prior work has demonstrated that LLMs can be used to simulate realistic student learning behaviors, including the development of learning curves and misconception patterns (Schmucker et al., 2024; Jin et al., 2024). Other studies simulate student behaviors conditioned on demographic, behavioral, and personality traits (Markel et al., 2023; Li et al., 2025a), as well as varying learning goals and trajectories (Sharma and Li, 2024).

6.2 Traditional Knowledge Tracing

Traditional KT models capture a student’s learning behavior in a sequence of her/his historical interactions, which consist of question tags, KC tags and binary responses (Chen et al., 2023; Liu et al., 2023a; Huang et al., 2023, 2026; Liu et al., 2025; Guo et al., 2026). For example, Piech et al. (2015) introduced the first DKT model, utilizing an LSTM to estimate knowledge mastery. Zhang et al. (2017) proposed a static key memory matrix to store KC relationships and a dynamic value memory matrix for predicting knowledge mastery levels. To incorporate the difficulty factor to enhance the knowledge state estimation, Ghosh et al. (2020) introduced a scalar difficulty parameter to control how far a question deviates from the underlying concept during question representation learning. Shen et al. (2022) established the relationship between student knowledge state and question difficulty level to improve KT performance. Liu et al. (2024) replaced question tags with difficulty levels to facilitate the translation of model predictions into interpretable

skill-level knowledge states, further highlighting the importance of difficulty modeling in KT.

7 Conclusion and Future Work

In this work, we introduce an interpretable difficulty-aware conversational KT framework that jointly models student ability and question difficulty in tutor-student dialogues. Our framework aims to capture fine-grained difficulty signals and provide more interpretable predictions of student performance, connecting LLM-based modeling with cognitively motivated assumptions. Quantitative results show modest improvements over existing KT baselines, suggesting the potential benefit of explicitly modeling both student ability and task difficulty. Empirical qualitative analyses provide further evidence for the interpretability of the learned parameters and indicate the value of incorporating an IRT predictor. In particular, the learned difficulty parameters show positive correlations with empirical difficulty statistics, suggesting that the model captures meaningful and cognitively plausible difficulty patterns across KCs. Additionally, the estimated ability trajectories often exhibit smooth growth patterns broadly consistent with gradual learning, while fluctuations in predicted correctness tend to correspond to changes in tutor-proposed task difficulty.

There are many avenues for future work. First, since we explicitly consider the difficulty factor in dialogue-based KT settings, we will further explore more effective methods for capturing fine-grained difficulty signals from tutoring dialogues and investigate how task difficulty evolves dynamically

during tutoring interactions. Second, for student knowledge state estimation, we plan to leverage advanced reasoning capabilities and reinforcement learning techniques to improve the accuracy and robustness of the estimation. In particular, future work will explore how multi-step reasoning can ensure consistency across dialogue turns and how reinforcement learning methods can refine estimation strategies based on long-term prediction performance. Third, we plan to explore misconception-aware conversational KT by detecting and tracking students' misconceptions from dialogue interactions. By integrating misconception signals with estimated knowledge states and difficulty parameters, the framework may provide more fine-grained and interpretable student performance prediction.

Acknowledgments

This work is partially supported by Renaissance Philanthropy under the learning engineering virtual institute (LEVI) initiative and the NSF under grant 2237676.

Limitations

There are several practical limitations to our work. First, since our work is the first to explicitly estimate the difficulty of tutor turns in dialogues, there are no existing difficulty prediction baselines we can compare against. Second, we only conduct experiments on math dialogues, and do not investigate other domains such as language learning or computer science. However, this is largely because KT in dialogues is currently only well-established in math. Finally, while our framework and other LLM-based approaches are significantly more accurate than traditional KT approaches, they are much more computationally expensive too. An important line of future work will be improving the efficiency of LLM-based approaches while maintaining high predictive accuracy.

Ethical Considerations

There are both potential societal benefits and risks associated with our work. Accurately modeling student ability and difficulty in dialogues is a necessary step to facilitating automated assessments in the dialogue setting. This can potentially benefit students through more accurate assessment and subsequently more targeted learning actions. However, as is the case with many other advances in educational AI, further automating assessments creates

a risk of replacing teachers' jobs. Our intention is for educational AI tools such as ours to be used in conjunction with teachers to improve outcomes for students, rather than be used to replace teachers. Finally, there is a risk of bias in our framework, as is common in all AI methods, due to inherent biases in training data. For instance, it is possible that students from populations that are less represented in training data are given less accurate assessments, therefore risking worse outcomes for those students. It is important that educational AI tools are thoroughly screened for bias before deployment with real students.

References

- John R Anderson. 2000. *Learning and memory: An integrated approach*. John Wiley & Sons.
- Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.
- Allan Birnbaum. 1968. Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*.
- James Dean Brown. 2013. Classical test theory. In *The Routledge handbook of language testing*, pages 337–349. Routledge.
- Carnegie Learning. 2024. Livehint overview. <https://support.carnegielearning.com/help-center/math/livehint/article/livehint-overview/>. Accessed: 2026-03-18.
- Jiahao Chen, Zitao Liu, Shuyan Huang, Qiongqiong Liu, and Weiqi Luo. 2023. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 14196–14204.
- Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. 2020. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the seventh ACM conference on learning@ scale*, pages 341–344.
- 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.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

- Teng Guo, Bojun Zhan, Shuyan Huang, Jiahao Chen, Xiangyu Zhao, Mingliang Hou, and Zitao Liu. 2026. Improving knowledge tracing through multi-source scaling with decoder-only transformers. *ACM Transactions on Intelligent Systems and Technology*, 17(3):1–25.
- Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Shuyan Huang, Zitao Liu, Qiongqiong Liu, Jiahao Chen, and Yaying Huang. 2026. Dual-attentional time-aware fusion networks for knowledge tracing. *Information Fusion*, 126:103573.
- Shuyan Huang, Zitao Liu, Xiangyu Zhao, Weiqi Luo, and Jian Weng. 2023. Towards robust knowledge tracing models via k-sparse attention. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 2441–2445.
- Hyounghook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach ai how to code: Using large language models as teachable agents for programming education. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–28.
- Khan Academy. 2023. Supercharge your teaching experience with khanmigo. <https://www.khanmigo.ai/>. Accessed: 2026-03-18.
- Jaewook Lee, Alexander Scarlatos, Simon Woodhead, and Andrew Lan. 2026. Letting tutor personas speak up for llms: Learning steering vectors from dialogue via preference optimization. *arXiv preprint arXiv:2602.07639*.
- Unggi Lee, Sungjun Yoon, Joon Seo Yun, Kyoungsoo Park, YoungHoon Jung, Damji Stratton, and Hyeoncheol Kim. 2024. Difficulty-focused contrastive learning for knowledge tracing with a large language model-based difficulty prediction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4891–4900.
- Haoxuan Li, Jifan Yu, Xin Cong, Yang Dang, Daniel Zhang-Li, Lu Mi, Yisi Zhan, Huiqin Liu, and Zhiyuan Liu. 2025a. Which type of students can llms act? investigating authentic simulation with graph-based human-ai collaborative system. *arXiv preprint arXiv:2502.11678*.
- Haoxuan Li, Jifan Yu, Yuanxin Ouyang, Zhuang Liu, Wenge Rong, Huiqin Liu, Juanzi Li, and Zhang Xiong. 2025b. Explainable few-shot knowledge tracing. *Frontiers of Digital Education*, 2(4):34.
- Guimei Liu, Huijing Zhan, and Jung-jae Kim. 2024. Question difficulty consistent knowledge tracing. In *Proceedings of the ACM web conference 2024*, pages 4239–4248.
- Zitao Liu, Shuyan Huang, Teng Guo, Mingliang Hou, and Qianru Liang. 2025. A prompt-driven framework for multi-domain knowledge tracing. *Machine Learning*, 114(4):87.
- Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Boyu Gao, Weiqi Luo, and Jian Weng. 2023a. Enhancing deep knowledge tracing with auxiliary tasks. In *Proceedings of the ACM web conference 2023*, pages 4178–4187.
- Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. 2023b. simplekt: a simple but tough-to-beat baseline for knowledge tracing. *The Eleventh International Conference on Learning Representations*.
- Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Jiliang Tang, and Weiqi Luo. 2022. pykt: a python library to benchmark deep learning based knowledge tracing models. *Advances in Neural Information Processing Systems*, 35:18542–18555.
- Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. Math-tutorbench: A benchmark for measuring open-ended pedagogical capabilities of llm tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 204–221.
- Julia M Markel, Steven G Opferman, James A Landay, and Chris Piech. 2023. Gpteach: Interactive training with gpt-based students. In *Proceedings of the tenth acm conference on learning@ scale*, pages 226–236.
- Allen Newell and Paul S Rosenbloom. 1981. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*, pages 1–55.
- Andre Nickow, Philip Oreopoulos, and Vincent Quan. 2020. The impressive effects of tutoring on prek-12 learning: A systematic review and meta-analysis of the experimental evidence.
- Benjamin D Nye, Arthur C Graesser, and Xiangen Hu. 2014. Autotutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24(4):427–469.
- Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems*, 28.
- Georg Rasch. 1993. *Probabilistic models for some intelligence and attainment tests*. ERIC.
- Frank E. Ritter and Lael J. Schooler. 2002. The learning curve. In *International Encyclopedia of the Social and Behavioral Sciences*. Pergamon.
- Alexander Scarlatos, Ryan S Baker, and Andrew Lan. 2025a. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 249–259.

- Alexander Scarlatos, Nigel Fernandez, Christopher Ormerod, Susan Lottridge, and Andrew Lan. 2025b. Smart: Simulated students aligned with item response theory for question difficulty prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25082–25105.
- Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. 2026. [Simulated students in tutoring dialogues: Substance or illusion?](#) Preprint, arXiv:2601.04025.
- Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025c. Training llm-based tutors to improve student learning outcomes in dialogues. In *International Conference on Artificial Intelligence in Education*, pages 251–266. Springer.
- Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 255–269.
- Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2024. Ruffle&riley: Insights from designing and evaluating a large language model-based conversational tutoring system. In *International Conference on Artificial Intelligence in Education*, pages 75–90. Springer.
- Paras Sharma and Qichang Li. 2024. Designing simulated students to emulate learner activity data in an open-ended learning environment. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 986–989.
- Shuanghong Shen, Zhenya Huang, Qi Liu, Yu Su, Shijin Wang, and Enhong Chen. 2022. Assessing student’s dynamic knowledge state by exploring the question difficulty effect. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 427–437.
- Yang Shi, Robin Schmucker, Min Chi, Tiffany Barnes, and Thomas Price. 2023. Kc-finder: Automated knowledge component discovery for programming problems.
- George S Snoddy. 1926. Learning and stability: a psychophysiological analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology*, 10(1):1.
- Jian Wang, Yinpei Dai, Yichi Zhang, Ziqiao Ma, Wenjie Li, and Joyce Chai. 2025. Training turn-by-turn verifiers for dialogue tutoring agents: The curious case of llms as your coding tutors. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12416–12436.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 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.
- Matthew Zent, Digory Smith, and Simon Woodhead. 2025. Piivot: A lightweight nlp anonymization framework for question-anchored tutoring dialogues. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27467–27476.
- 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 Prompts

A.1 Prompt for Knowledge Estimation

You are an experienced math teacher. Given a dialogue where the student is working through a multiple-choice diagnostic math question, based on the student’s responses and demonstrated understanding in the dialogue, classify the student’s current ability level. Respond with exactly one token from: GOOD BAD.

[BEGIN QUESTION]
Question Text
[END QUESTION]

[BEGIN DIALOGUE]
Dialogue Context
[END DIALOGUE]

Specifically, the input prompt I^K consists of the following three key components:

- **Task Definition:** A natural language instruction that defines the task and specifies the expected output format for the LLM.
- **Question Text:** The diagnostic question presented to the student, typically enclosed within structured markers [BEGIN QUESTION] and [END QUESTION].
- **Dialogue Context:** The historical teacher–student interaction associated with the question, enclosed within markers [BEGIN DIALOGUE] and [END DIALOGUE].

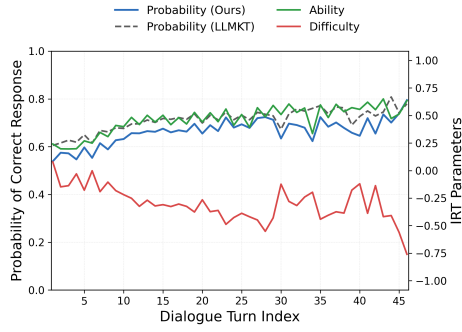


Figure 5: Learning trajectories for dialogues in the QATD_{2k} dataset.

A.2 Prompt for Difficulty Estimation

You are an experienced math teacher. Given a dialogue where the student is working through a multiple-choice diagnostic math question, based on the question content and required knowledge components, classify the question’s difficulty level. Respond with exactly one token from: HARD EASY.

[BEGIN QUESTION]
Question Text
[END QUESTION]

[BEGIN DIALOGUE]
Dialogue Context
[END DIALOGUE]

[BEGIN CURRENT TEACHER TURN]
Teacher-posed Task
[END CURRENT TEACHER TURN]

[BEGIN KC]
KC Names
[END KC]

Specifically, the input prompt I^D consists of the following two key components:

- **Teacher-posed Task:** The next teacher-posed question that the student is expected to respond to is enclosed within markers [BEGIN CURRENT TEACHER TURN] and [END CURRENT TEACHER TURN], representing the immediate task used for predicting the student’s knowledge state.
- **KC Names:** The associated KCs with the next tutor-posed task enclosed within markers [BEGIN KC] and [END KC]. The model must infer

from prior interactions whether the student has mastered the required knowledge.

B Additional “Learning Curve” Analysis

We provide learning curve analysis based on dialogue level in Figure 5.

Rubrics as Semantic Subspaces: A Unified Approach to Rubric-based Constructed Response Scoring across Short Answers and Essays

Sebastian Gombert¹, Sonja Hahn¹, Nico Andersen¹, Leon Camus¹, Zhifan Sun¹,
Ngoc Nhu Hao Nguyen¹, Fabian Zehner^{1,2}, Longwei Cong¹, Alexander Mehler⁴
Hendrik Drachsler^{1,3,4}

¹DIPF | Leibniz-Institute for Research and Information in Education

²Centre for International Student Assessment (ZIB)

³Studiumdigitale & ⁴Computer Science Department, Goethe University Frankfurt

Abstract

Rubrics are the primary reference for manual scoring of constructed responses, and there is growing interest in their use in automated scoring methodologies. In this work, we propose Aspect-Grounded Rubric-Answer Alignment (AGRAA), a rubric-based end-to-end scoring framework that models rubric descriptors as latent aspect spaces. Concretely, rubric descriptors are represented as low-dimensional subspaces derived from contextualised transformer embeddings, and student responses are scored according to how strongly their representations align with these rubric-induced spaces relative to the residual space outside them. This formulation provides a geometrically grounded interpretation of rubric-based scoring while enabling end-to-end training with standard transformer encoders. We introduce three distinct architectural variants and evaluate them on multiple short-answer and essay scoring datasets. Across these tasks, AGRAA achieves predictive performance highly competitive with strong neural and feature-based baselines. In addition, the framework yields interpretable intermediate representations that expose which rubric-defined aspects contribute to scoring decisions, enabling decision-aligned explanations grounded in rubric descriptors.

1 Introduction

Rubric-based scoring plays a central role in assessing constructed responses in educational settings. Rubrics describe performance using qualitative criteria that enable fine-grained evaluation of responses (Reddy and Andrade, 2010). As automated constructed-response assessment becomes increasingly important in digital learning environments (Mello et al., 2024) and large-scale testing (Zehner et al., 2025), there is growing interest in methods that can automatically score responses while leveraging the criteria defined in the rubrics (Sonkar et al., 2024; Wang et al., 2019; Gombert

et al., 2026; Cong et al., 2026; Sreedevi et al., 2026; Eltanbouly et al., 2025).

Across domains and response types, rubrics typically describe performance in terms of aspects such as conceptual correctness, completeness, use of evidence, or argumentative quality (Reddy and Andrade, 2010). These aspects are usually distributed across the response rather than localised to individual words or spans. For example, a science rubric might require identifying oxidation, explaining oxygen exposure, and linking the process to the observed color change in an apple. Scoring therefore amounts to estimating how strongly a response expresses the aspects associated with each rubric level.

In this work, we frame rubric-based scoring as a form of *neural subspace classification* (Fukunaga, 2013; Fukui, 2021) on top of contextual embeddings produced by transformer encoder models. Each rubric level defines a low-dimensional semantic subspace derived from the dominant directions of variation in its textual representation. These directions can be interpreted as latent aspects of the rubric. Student responses are projected into these rubric-induced spaces, and scoring is implemented by measuring how much of the response representation lies within each rubric subspace relative to the residual space outside it. The summarised contributions of this paper are:

- We introduce **AGRAA**, a rubric-based representation learning framework that aligns student responses with rubric descriptors through subspace-based similarity, enabling rubric-aware scoring decisions that expands upon earlier ideas introduced by Gombert et al. (2026).
- We propose three architectural variants: a *bi-encoder* formulation (Bi-AGRAA), a *level-wise cross-encoder* formulation (Cross-AGRAA), and a *joint cross-encoder* formulation (Joint-Cross-AGRAA) that allows direct

interaction between rubric levels during representation learning.

- Through experiments on multiple short-answer and essay scoring datasets, we demonstrate that rubric-based subspace alignment achieves **competitive predictive performance** while providing transparent, rubric-aligned explanations of model decisions.
- We show that the approach yields **interpretable intermediate representations** that reveal which aspects within rubric descriptors contribute to a prediction.

2 Background

2.1 Automated Assessment of Constructed Responses

The automated assessment of constructed responses is a long-established research branch in educational natural language processing (Bai and Stede, 2022; Burrows et al., 2014; Bexte et al., 2024). In prior research, two genres of constructed responses, namely essays and short answers, were the main research focus, with distinct research trajectories for both genres (Bai and Stede, 2022). While short answer scoring mainly focuses on the content of given answers ("content scoring") (Bexte et al., 2024; Ziai et al., 2012), essay scoring is mainly focused on writing-level traits such as coherence, grammar, argumentation, or style (Bai and Stede, 2022), although there is some work that instead deals with essay content scoring (Gombert et al., 2024).

2.2 Rubrics in Automated Short Answer Scoring (ASAS)

Scoring rubrics have previously been explored as inputs to automatic short-answer scoring systems. Wang et al. (2019) proposed a BiLSTM-based architecture that computes word-level alignment scores between tokens in a student response and tokens in rubric descriptors. They showed that incorporating rubric information improves performance compared to a plain BiLSTM-based scoring model introduced by Riordan et al. (2017), particularly in low-resource settings. These results provide early evidence that explicit modeling of rubric information can benefit automatic short-answer scoring.

Gombert et al. (2026) proposed two architectures for rubric-based short answer scoring. Both architectures use attention to calculate alignment

scores between responses and rubric descriptors using embeddings from transformer encoder models. The performance level whose corresponding rubric has the highest alignment to a given response is assigned as the final score. The variants differ in how alignment is calculated, on top of mean-pooled span embeddings or between the embeddings of individual rubric descriptor and response tokens.

Cong et al. (2026) and Wei et al. (2025) evaluated the utility of rubrics for short answer scoring when using LLM in-context learning. In both cases, providing the LLM with rubrics significantly improved performance compared to a baseline without rubrics and was more beneficial than providing simple few-shot examples.

2.3 Rubrics in Automated Essay Scoring

Sreedevi et al. (2026) proposed a rubric-aware neural architecture for automated essay scoring that explicitly incorporates analytic rubric criteria into the modeling process. Essays are encoded using contextual embeddings augmented with discourse information. These discourse-aware essay representations are then connected to rubric traits through trait-specific attention mechanisms, enabling the model to learn rubric-conditioned representations of essay quality. The model jointly predicts both holistic essay scores and rubric-level trait scores, with the rubric acting as a structured supervisory signal that guides the decomposition of essay quality into interpretable evaluation dimensions.

Eltanbouly et al. (2025) implemented an essay scoring framework that uses LLMs to extract features from given rubrics that are then used in a feature-based statistical classifier. Concretely, they prompt LLMs to transform given rubrics into binary yes-no questions that ask whether a given rubric-defined aspect is fulfilled by the essay or not. In the next step, the LLM is prompted to answer these individual questions. The resulting binary features are used as input features for a logistic regression classifier.

3 Aspect-Grounded Rubric-Answer Alignment (AGRAA)

Rubrics typically describe the quality of the answer in terms of qualitative criteria such as conceptual correctness, completeness, use of evidence, or quality of the explanation (Reddy and Andrade, 2010; Popham, 1997). Importantly, these descriptions rarely correspond to a single lexical cue or an iso-

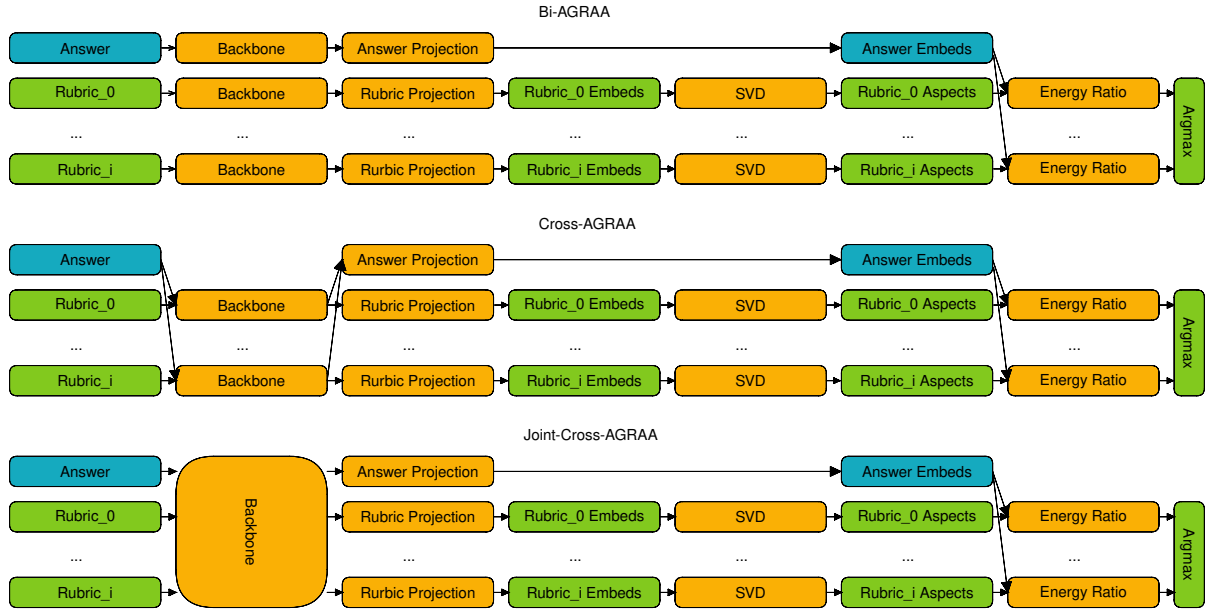


Figure 1: A schematic depiction of both architectures. SVD = Singular Value Decomposition. Questions are included as context in each forward pass.

lated fact. Instead, rubric levels implicitly refer to *aspects* that characterise responses at different quality levels (Popham, 1997). For example, a “correct” rubric level in a science task may simultaneously require the identification of a process, explanation of a mechanism, and appropriate use of domain terminology. These aspects are distributed across the response (Popham, 1997) and can be realised through a wide range of linguistic expressions (Kreidler, 1998).

Student responses also do not express rubric-relevant aspects in a single location. Rather, aspects are typically distributed throughout the text and expressed with varying degrees of completeness and clarity. We therefore can interpret both rubric levels and student responses as *configurations of aspects*. Each rubric level defines a set of aspects expected for that level, while each response expresses these aspects to varying degrees. The central task of rubric-based scoring can thus be framed as estimating how strongly a response expresses the aspects associated with each rubric level. Therefore, our approach is based on three core assumptions:

1. Each rubric level descriptor defines a set of latent aspects.
2. The responses express these aspects with varying strength.
3. Scoring corresponds to measuring the align-

ment between the aspects defined in the rubric and their materialisation in a given response.

Based on these assumptions, we implemented *AGRAA*, an architectural framework for end-to-end scoring of constructed responses grounded in subspace classification methodology (Fukui, 2021).

3.1 Encoder and Asymmetric Projection

Given a question/item q and a response a , we aim to predict the correct performance level $0, \dots, i$ for answer a based on the qualitative criteria described in rubric R consisting of performance level-wise descriptors $r_0, \dots, r_i$. For this purpose, we consider three architectural variants that differ in how contextual interactions between the response and the rubric are modeled. Figure 1 depicts these variants schematically.

Bi-AGRAA. The first variant is a *bi-encoder* setup (*Bi-AGRAA*), where the response and each rubric level are encoded independently. In this case, the response representation is shared across all rubric levels. The question is added as contextual input at each forward pass but masked during all downstream processing steps.

Cross-AGRAA The second variant is a *level-wise cross-encoder* setup (*Cross-AGRAA*), where the encoder jointly processes the response together with a single rubric level at a time. This produces response representations that are conditioned on the

rubric level under consideration. The question is added as contextual input at each forward pass but masked during all downstream processing steps.

Joint-Cross-AGRAA The third variant is a *joint cross-encoder* setup (*Joint-Cross-AGRAA*), where the question, response, and all rubric levels are encoded together in a single forward pass. This allows for direct contextual interaction between rubric levels while producing a single globally contextualised response representation. The question is added as contextual input at each forward pass, but is masked for all downstream processing steps. This is the exact setup used by [Gombert et al. \(2026\)](#) for their architectures.

In all variants, functional tokens such as CLS or SEP are masked for all following computations to prevent the models from learning uninterpretable shortcuts involving these tokens. Moreover, all tokens of q are masked for all steps after encoding. Each of the three variants yields contextualised embeddings for response and rubric tokens:

$$H_a \in \mathbb{R}^{T_a \times d}, \quad H_{r_i} \in \mathbb{R}^{T_{r_i} \times d} \quad (1)$$

In these, d denotes the encoder hidden size. To account for the different functional roles of rubric and response text (the former describes in which aspects a correct response materialises, while the latter is a concrete materialisation), we apply asymmetric linear projections into a shared comparison space of dimension l :

$$X_a = H_a W_a + B_a, \quad X_a \in \mathbb{R}^{n_a \times l} \quad (2)$$

$$X_{r_i} = H_{r_i} W_r + B_r, \quad X_{r_i} \in \mathbb{R}^{n_{r_i} \times l} \quad (3)$$

Separate projection layers allow rubric tokens to define aspect directions while response tokens are evaluated relative to these directions.

3.2 Rubric-induced Aspect Spaces

On top of the three underlying encoding approaches, we use the same specialised head, which can be viewed as a neural subspace classifier ([Fukunaga, 2013](#)) in which each rubric level descriptor defines a semantic subspace of one or more latent aspects. In this perspective, scoring corresponds to a form of nearest-subspace classification where responses are assigned to the rubric level whose semantic subspace best explains their representation. We estimate these aspects by constructing a low-dimensional subspace from the rubric token embeddings. Given projected rubric token embeddings

X_{r_i} , we compute a singular value decomposition:

$$X_{r_i} = U_i \Sigma_i V_i^\top \quad (4)$$

The right singular vectors define orthogonal directions in the projection space capturing dominant semantic variation within the rubric text. We interpret these directions as latent aspects. To determine the number of aspects a given rubric descriptor contains, which of course differs from descriptor to descriptor, we retain the smallest number of singular directions explaining a fixed proportion of variance τ (e.g., 0.9), capped by a maximum number of aspects $k_{\max}$. Both need to be set as hyperparameters depending on the granularity of the rubrics at hand:

$$k_i = \min\left(\{k_{\max}\} \cup \{k : \frac{\sum_{j=1}^k \sigma_{ij}^2}{\sum_j \sigma_{ij}^2} \geq \tau\}\right) \quad (5)$$

This yields an orthonormal basis $Q_i \in \mathbb{R}^{l \times k_i}$ representing the aspect space associated with rubric level i , together with singular values σ_{ij} .

3.3 Response-to-Aspect Alignment

To evaluate how strongly a response expresses each rubric aspect, we project response token embeddings into the rubric-induced subspace:

$$A_i = X_a Q_i \in \mathbb{R}^{n_a \times k_i} \quad (6)$$

Each column of A_i represents the strength with which the response tokens align with a particular rubric aspect. To quantify how well a response can be expressed within the rubric-induced aspect space, we compute the proportion of response variance explained by this subspace. Let

$$E_{\parallel,i} = \frac{1}{n_a} \|A_i\|_F^2 \quad (7)$$

denote the energy of the response within the rubric-induced subspace, and

$$E_{\text{tot}} = \frac{1}{n_a} \|X_a\|_F^2 \quad (8)$$

The total response energy in projection space. The residual energy outside the rubric subspace is:

$$E_{\perp,i} = E_{\text{tot}} - E_{\parallel,i}. \quad (9)$$

We then define the alignment between a response and rubric level i as the log-ratio between the energy explained by the rubric-induced subspace and the residual energy outside the subspace, a form closely related to the classical subspace classification criteria in statistical pattern recognition ([Fukunaga, 2013](#)):

$$s_i = \log \frac{E_{\parallel,i} + \varepsilon}{E_{\perp,i} + \varepsilon}. \quad (10)$$

3.4 Classification and Training

The predicted scores are obtained by an argmax in all $s_0, \dots, s_i$:

$$\hat{y} = \arg \max_i s_i. \quad (11)$$

The model is trained end-to-end using softmax cross-entropy loss over these alignment scores.

Aspect distribution regularisation. While the above objective optimises predictive performance, early pre-experiments showed that the resulting models often collapsed onto one dominant “master” aspect that largely determined correctness, while the remaining directions acted as residual aspects without a clear function. This constitutes a learned shortcut. However, as stated earlier, rubrics typically describe performance across multiple aspects. To encourage the model to better reflect this multi-aspect structure, we introduce an additional regularisation term to the loss function that promotes a more distributed use of rubric-based semantic directions.

To obtain a direction-level summary of how strongly each aspect contributes to the alignment with a rubric unit, we first compute pooled per-direction alignment values:

$$e_{i,j} = \sqrt{\frac{1}{\alpha} \ln \sum_{t=1}^{n_a} \exp(\alpha (A_i)_{t,j}^2)}. \quad (12)$$

These values summarise the strongest token-level evidence for each semantic direction. We then derive a normalised distribution over aspect contributions

$$p_k = \frac{e_{i,k}}{\sum_{m=1}^{k_i} e_{i,m}}, \quad (13)$$

and compute its entropy

$$\mathcal{H}(p) = - \sum_{k=1}^{k_i} p_k \log p_k. \quad (14)$$

In practice, this entropy is normalised with respect to the number of retained directions and is used as a soft constraint: the model is penalised only when the aspect distribution becomes too concentrated, that is, when its entropy falls below a predefined target range. This encourages the model to make use of multiple rubric-defined semantic directions without forcing all directions to contribute equally. Because regularisation operates only on the dynamically retained directions for a given rubric unit

through variance-based subspace construction, it promotes a balanced use of the available semantic directions without forcing the model to use the maximum possible number of aspects in every case. The final loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{ent}}, \quad (15)$$

where $\mathcal{L}_{\text{ent}}$ denotes the entropy-based penalty and λ controls its strength.

3.5 Interpretability

To obtain decision-aligned explanations, we decompose the final prediction into contributions of individual aspects. Let $e_{i,j}$ denote the pooled response energy along the aspect j of rubric level i (as defined in equation 12), and $w_{i,j}$ the corresponding normalised singular-value weight. Within each rubric level, aspect contributions are computed as:

$$\tilde{c}_{i,j} = \frac{w_{i,j} e_{i,j}^2}{\sum_{m=1}^{k_i} w_{i,m} e_{i,m}^2}. \quad (16)$$

To align the explanation with the model’s cross-level decision, these quantities are weighted by softmax-normalised level scores:

$$c_{i,j} = \text{softmax}(s)_i \cdot \tilde{c}_{i,j}. \quad (17)$$

The values $c_{i,j}$ sum to one across all levels and aspects and therefore directly quantify how much each rubric-defined semantic direction contributed to the final classification. The resulting visualisations make explicit which rubric aspects drove the decision and which rubric tokens characterise these aspects (see Figure 2). This yields a decision-aligned explanation in terms of rubric semantics. Because explanations are derived directly from the same projected response representations used in the scoring function, the resulting aspect contributions reflect the internal evidence underlying the model’s decision rather than a post-hoc approximation.

3.6 Applicability across response types

Given a long-context encoder is used as backbone, the architecture can support constructed response types of varying length. Short answers and essays, as the most important types of written constructed responses, differ primarily in the number, focus, and complexity of aspects expressed, as well as in overall length. This can be accommodated by adjusting the maximum number of retained aspect directions k_{max} .

Approach	UA P	UA R	UA F1	UA QWK	UQ P	UQ R	UQ F1	UQ QWK
ModernGBERT-1B Joint-Cross-AGRAA	.760	.740	.743	.748	.620	.577	.581	.539
ModernGBERT-1B Cross-AGRAA	.745	.734	.736	.733	.594	.584	.586	.521
ModernGBERT-1B Bi-AGRAA	.726	.711	.714	.705	.567	.552	.558	.475
ModernGBERT-1B Classifier (Q+A+Ru)	.732	.716	.719	.713	.579	.553	.553	.466
ModernGBERT-1B Classifier (Q+A)	.707	.704	.705	.692	.533	.529	.522	.404
GBERT-large ToLeGRAA (Gombert et al., 2026)	.748	.739	.742	.751	.620	.595	.600	.569
Claude Sonnet 4.5 5-shot (Gombert et al., 2026)	.680	.679	.679	.665	.646	.640	.641	.603

Table 1: Results achieved for the ALICE-LP dataset. We follow the evaluation protocol established in Gombert et al. (2026) and report dataset-wide metrics for the unseen answers and unseen questions evaluation sets. P = Weighted Precision. R = Weighted Recall. F1 = Weighted F1. QWK = Quadratic Weighted Kappa. UA = Unseen Answers subset. UQ = Unseen Questions subset.

4 Evaluation

Following the definitions from Murdoch et al. (2019), we evaluate the models for both *predictive* and *descriptive* accuracy. While predictive accuracy refers to the predictive performance of models, descriptive accuracy concerns whether these models make predictions for plausible reasons, i.e., whether plausible aspects are extracted from the rubrics and whether the calculated alignment matches what is stated in a given input answer. Our evaluation, therefore, addresses two concrete research questions:

RQ1. What is the *predictive accuracy* of the three AGRAA variants?

RQ2. Are the models faithful in terms of *descriptive accuracy* in regard to the content of answers and rubrics?

For the first research question, we measure prediction quality, while for the second research question, we qualitatively analyzed 64 randomly selected examples from the unseen-answers evaluation set of the *ALICE-LP* dataset (see next section) for each model variant (Bi-AGRAA, Cross-AGRAA, Joint-Cross-AGRAA).

4.1 Datasets

For our experiments, we used two datasets for short-answer scoring and one for holistic essay scoring, namely *ALICE-LP*, *ASAP-SAS*, and *PERSUADE 2.0*. In every evaluation, the models are trained on the whole respective dataset.

ALICE-LP 1.0. This short answer scoring dataset was first introduced by Gombert et al. (2026) and includes 13,075 German short answers to 118 questions from four different STEM domains collected in German middle and high schools. All answers are rated on a three-point

scale following question-specific rubrics. It is divided into a test and two evaluation sets. One of these contains answers to questions seen during training (*unseen answers*), while the other contains answers to questions unseen during training (*unseen questions*).

ASAP-SAS. This short-answer scoring dataset was first introduced in the context of a 2012 Kaggle competition (Barbara et al., 2012) and has since grown into a widely used benchmark for automated short-answer scoring. It comprises 22,067 English answers to a total of ten different questions with question-specific rubrics, including both STEM and reading comprehension questions, collected at US-American schools.

PERSUADE 2.0. This essay scoring dataset was first introduced by Crossley et al. (2024). It comprises 25,996 argumentative essays written in English for 15 different prompts. It includes holistic scores on a 6-point scale, as well as analytic scores for multiple argumentation-related criteria on a 3-point scale. In this work, we focus on predicting the holistic scores. The dataset contains source-based and independent writing prompts, both of which are scored using slightly different rubrics.

4.2 Backbone Models, Hyperparameters and Baselines

As backbone models, we used *ModernBERT* (Warner et al., 2025) variants. *ModernBERT* is a modified variant of the original *BERT* architecture that supports long-context processing with the help of RoPE embeddings (Su et al., 2024). For the German *ALICE-LP* set, we use *ModernGBERT-1B* (Wunderle et al., 2025), a German-specific *ModernBERT* (Warner et al., 2025) variant, while for the English *ASAP-SAS* and *PERSUADE 2.0*, we use the regular *ModernBERT-large* (Warner et al., 2025). As baselines, we employ two classifier vari-

Approach	1	2	3	4	5	6	7	8	9	10	QWK _m	QWK _F
ModernBERT-large Joint-Cross-AGRAA	.871	.883	.709	.760	.796	.887	.714	.667	.830	.782	.790	.802
ModernBERT-large Cross-AGRAA	.874	.878	.715	.717	.810	.883	.732	.664	.846	.749	.786	.800
ModernBERT-large Bi-AGRAA	.885	.863	.706	.748	.837	.835	.706	.679	.842	.740	.784	.795
ModernBERT-large Classifier (Q+A)	.857	.831	.729	.707	.802	.806	.716	.658	.825	.759	.769	.776
ModernBERT-large Classifier (Q+A+Ru)	.859	.816	.654	.730	.791	.803	.732	.641	.829	.783	.764	.772
Ensemble of fine-tuned MLMs (Ormerod, 2022)	.882	.868	.722	.750	.813	.822	.734	.702	.865	.779	.796	.803
Random Forests + div. feats. (Kumar et al., 2019)	.872	.824	.745	.743	.845	.858	.725	.624	.843	.832	.791	.802
GTE-EN-MLM Classifier (Q+A+R) (Gombert et al., 2026)	.868	.848	.715	.738	.790	.827	.681	.718	.844	.773	.780	.788
ModernBERT-large GRAASP (Gombert et al., 2026)	.876	.852	.724	.727	.803	.871	.702	.672	.802	.749	.779	.788
GRU (Riordan et al., 2019)	.830	.791	.662	.731	.844	.861	.736	.664	.809	.777	.771	.779
ModernBERT-large Classifier (Q+A+Ru) (Gombert et al., 2026)	.856	.836	.677	.703	.768	.831	.699	.675	.817	.748	.761	.775

Table 2: Results achieved for the ASAP-SAS dataset. We follow the evaluation protocol used by Riordan et al. (2019), Kumar et al. (2019), Gombert et al. (2026), and Ormerod (2022) to report question-wise Quadratic Weighted Kappa scores, which are then averaged using the arithmetic (QWK_m) as well as the Fisher-weighted (QWK_F) mean.

ants of these models. For the first variant, a given classifier receives a response and the corresponding question/item prompt, and for the second variant, the classifier is also given the full rubrics. For *ALICE-LP* and *ASAP-SAS*, we set $k_{\max} = 3$, and for *PERSUADE 2.0*, we set $k_{\max} = 12$, based on a manual inspection of the number of aspects contained in the corresponding rubric descriptors. We set $\tau = 0.9$, $\lambda = 0.2$, and $\alpha = 5.0$ in all cases. Training was run for 12 epochs with a learning rate of $1e-5$, a weight decay of 0.1, gradient clipping set to 1.0, a batch size of 16, and early stopping active using *AdamW* as optimiser. We report results for the best checkpoint per run.

5 Results

5.1 Predictive Accuracy

Across all evaluated datasets, the proposed AGRAA variants achieve competitive predictive performance. When compared against conventional classifier architectures built on the same underlying backbone models, several AGRAA variants consistently match or outperform these baselines. On the *ALICE-LP* dataset, *Joint-Cross-AGRAA* reaches performance comparable to the previously reported *ToLeGRAA* model on the unseen answers setting, while all AGRAA variants outperform both rubric-aware and rubric-free classifier baselines implemented with the same encoder. On the *ASAP-SAS* dataset, the AGRAA variants likewise improve over the corresponding classifier baselines and achieve results competitive with strong published neural and feature-based systems, with *Joint-Cross-AGRAA* and *Cross-AGRAA* closely approaching the best reported aggregate QWK scores. On the essay scoring dataset, the proposed models perform on par with or slightly

Approach	P	R	F1	QWK
ModernBERT-large Cross-AGRAA	.694	.693	.691	.878
ModernBERT-large Joint-Cross-AGRAA	.690	.689	.688	.873
ModernBERT-large Bi-AGRAA	.672	.668	.666	.866
ModernBERT-large Class. (Q+A+Ru)	.681	.679	.675	.869
ModernBERT-large Class. (Q+A)	.668	.667	.664	.864
DeBERTa-v3 Class. (Ravindran and Choi, 2025)	-	-	-	.870
ModernBERT Class. (Ormerod, 2025)	-	-	-	.868

Table 3: Results achieved for the *PERSUADE 2.0* dataset. We report weighted precision (P), weighted recall (R), weighted F1 score, and quadratic weighted Kappa.

above strong transformer-based baselines.

Taken together, these results indicate that the rubric-based subspace formulation underlying AGRAA can match, and, in several cases, exceed the predictive performance of conventional classifier formulations using the same encoder backbones.

5.2 Descriptive Accuracy

Overall, the Bi-AGRAA and Joint-Cross-AGRAA variants frequently decomposed rubric representations into several semantically meaningful aspects that corresponded to distinct rubric criteria or conceptual components.

Across most inspected visualisations, the extracted aspects highlighted meaningful content words from the rubric descriptors as well as modifiers associated with them, such as determiners or descriptive adjectives (e.g., *die*, *der*). This suggests that the learned semantic directions are anchored in linguistically and semantically plausible parts of the rubric descriptors.

Figure 2 illustrates an explanation produced by the *Joint-Cross-AGRAA* model. Within the predicted rubric level, three aspects that characterise the expected reasoning are visible. The

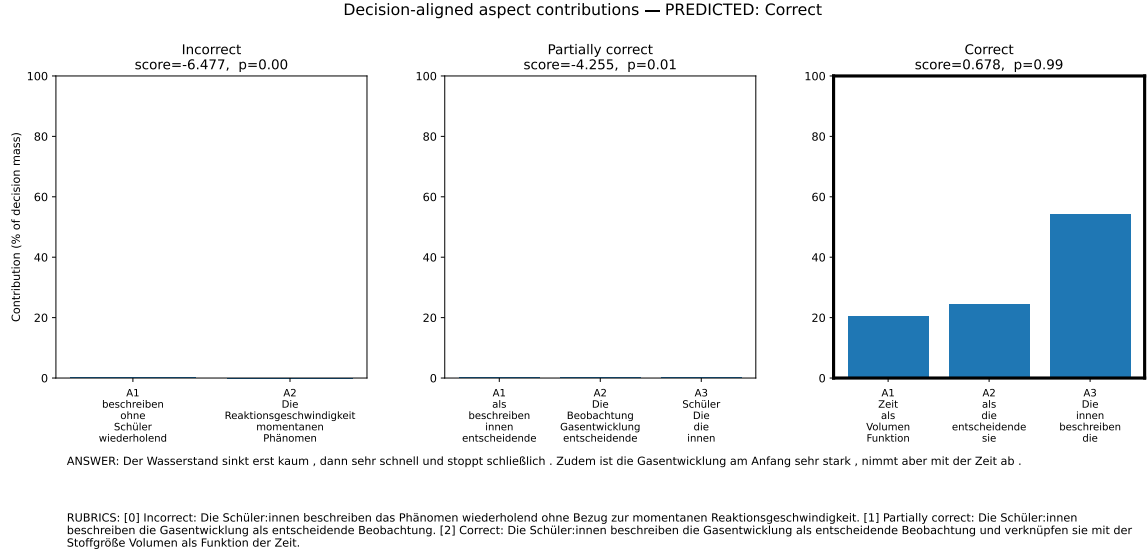


Figure 2: This figure shows an example of the aspect contributions with a correct answer from the ALICE-LP dataset the system rightfully judged as correct. Each bar represents the $c_{i,j}$ (see Equation 17) for a given score s_i (see Equation 10) and a given aspect j for a Cross-AGRAA *ModernGBERT*-1B model trained with $k_{max} = 3$. Since the model is highly confident, there are only bars visible in the correct cell. For each aspect, we show the four most characteristic words from the respective rubric.

most influential aspect is associated with the token *beschreiben*, reflecting the rubric requirement that the response should provide a description. A second aspect is characterised by tokens such as *Zeit*, *Volumen*, and *Funktion*, corresponding to the rubric requirement that the gas volume should be interpreted as a function of time. A third aspect reflects observational terminology such as the adjective *entscheidende* in *Beobachtung*, capturing the emphasis on identifying the gas development as the key phenomenon. The student response indeed describes how the gas development changes over time (“*Gasentwicklung ... nimmt aber mit der Zeit ab*”), which aligns closely with these rubric-defined aspects, suggesting that the extracted directions correspond to semantically meaningful components of the rubric. Across all the inspected examples, we could observe similar patterns of align

dicating that cross-attention between rubric descriptors as well as between rubric descriptors and answers is needed to best differentiate between performance levels. Overall, we can conclude that neural subspace classification as employed for **AGRAA** can be seen as beneficial for predictive performance in rubric-based constructed response scoring.

Moreover, the subspace formulation yields interpretable intermediate representations that expose how rubric-defined aspects contribute to scoring decisions. Qualitative analysis indicated that these aspects frequently correspond to meaningful rubric aspects, suggesting that the learned semantic directions capture pedagogically relevant structure present in the rubrics. However, it needs to be confirmed whether this pattern holds up across varied models and datasets, which needs to be the objective of future work.

6 Discussion

The experimental results suggest that rubric-based scoring can be effectively framed as a subspace alignment problem between responses and rubric descriptors. Across three datasets covering short answer- and essay scoring, the proposed **AGRAA** variants achieved predictive performance stronger than plain transformer classifier baselines in the majority of cases. The strongest among these across all datasets is the **Joint-Cross-AGRAA** variant, in-

Limitations

Several limitations remain. First, the interpretability analysis in this work is primarily qualitative and future work could investigate quantitative measures of explanation quality. Second, the number of retained semantic directions is controlled through global hyperparameters, which may not always perfectly reflect the conceptual granularity of a given rubric. We did not conduct an extensive hyperparameter search. A small-scale stability analysis (see

Appendix) revealed that the predictive performance stays stable under varied hyperparameters, but may experience drops for certain configurations. Finally, the approach assumes that rubric descriptors sufficiently capture the semantic structure of the scoring task, which may not hold for poorly specified rubrics. Therefore, future work would need to test the relationship between variables such as linguistic and conceptual uncertainty within rubrics and the resulting predictive performance.

Ethical Statement

Automatic constructed response scoring is an educational NLP task. The EU AI act (European Parliament and Council of the European Union, 2024) labels AI technology (including NLP technology) in education rightfully as a high-risk application. While the individual risk depends highly on the exact context in which the corresponding technology is used and must be assessed case-by-case, mispredictions can tremendously impact learner success even in low-stakes scenarios.

For example, there is clear empirical evidence that negative feedback (and the predicted performance levels, if low, are nothing but that, if presented to a given learner) can hurt the intrinsic motivation of learners (Fong et al., 2019). If a system based on one of our presented approaches wrongly scores correct answers as wrong, learner motivation might thus unnecessarily suffer. Even worse, when such mistakes happen in high-stakes assessments, it might negatively affect students' overall life path since, in many countries, access to university programs and jobs is highly coupled with assessment results, e.g., in the form of GPA scores. Deployment in such scenarios, therefore, requires extensive evaluation.

On the other hand, if a model is, for example, used in formative assessment and mispredicts a given wrong student answer as being correct, the corresponding student might not revise possible misconceptions present in their answer. If this happens too often throughout a given unit, students might develop misunderstandings about the content. Moreover, there is already existing research on teacher dashboards that comprehensively summarise student performance so teachers can plan interventions based on that (Karademir et al., 2024). If a non-reliable short-answer scoring system powers such a dashboard, teachers might make incorrect interventions, which, in turn, could harm stu-

dent learning.

Another aspect that needs to be further assessed, which was out of the scope of this particular study, is whether the underlying models replicate undesired biases. An example of this might be a possible bias against students with dysgraphia or dyslexia. If dyslexic or dysgraphic writing is not sufficiently represented in a given training set, systems might encounter problems dealing with the same, hurting downstream predictive performance for student answers formulated by affected students.

References

- Xiaoyu Bai and Manfred Stede. 2022. A survey of current machine learning approaches to student free-text evaluation for intelligent tutoring. *International Journal of Artificial Intelligence in Education*, 33(4):992–1030.
- Barbara, Ben Hamner, Jaison Morgan, lynnvandev, and Mark Shermis. 2012. The hewlett foundation: Short answer scoring. <https://kaggle.com/competitions/asap-sas>. Kaggle.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2024. Strengths and weaknesses of automated scoring of free-text student answers. *Informatik Spektrum*, 47(3–4):78–86.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2014. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1):60–117.
- Longwei Cong, Leon Hammerla, Sonja Hahn, Sebastian Gombert, Hendrik Drachsler, and Ulf Kröhne. 2026. Automatic short answer grading with llms: From memorization to reasoning. In *Proceedings of the 16th International Learning Analytics and Knowledge Conference (LAK '26)*, New York, NY, USA. ACM.
- Scott Andrew Crossley, Yu Tian, Perpetual Baffour, A. Franklin, M. Benner, and U. Boser. 2024. A large-scale corpus for assessing written argumentation: Persuade 2.0. *Assessing Writing*, 61:100865.
- Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. TRATES: Trait-specific rubric-assisted cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. Association for Computational Linguistics.
- European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and

- (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act). <https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng>. OJ L 2024/1689, 12 July 2024.
- Carlton J Fong, Erika A Patall, Ariana C Vasquez, and Sandra Stautberg. 2019. A meta-analysis of negative feedback on intrinsic motivation. *Educational Psychology Review*, 31(1):121–162.
- Kazuhiro Fukui. 2021. Subspace methods. In *Computer vision: a reference guide*, pages 1221–1224. Springer.
- Keinosuke Fukunaga. 2013. *Introduction to statistical pattern recognition*. Elsevier.
- Sebastian Gombert, Aron Fink, Tornike Giorgashvili, Ioana Jivet, Daniele Di Mitri, Jane Yau, Andreas Frey, and Hendrik Drachler. 2024. [From the automated assessment of student essay content to highly informative feedback: a case study](#). *International Journal of Artificial Intelligence in Education*, 34(4):1378–1416.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berit Katharina Czinczel, David Bednorz, Marcus Kubsch, Daniele Di Mitri, Knut Neumann, and Hendrik Drachler. 2026. Are rubrics all you need? towards rubric-based automatic short answer scoring via guided rubric–answer alignment. In *Proceedings of the 16th International Learning Analytics and Knowledge Conference*.
- Onur Karademir, Lena Borgards, Daniele Di Mitri, Sebastian Strauß, Marcus Kubsch, Markus Brobeil, Adrian Grimm, Sebastian Gombert, Nikol Rummel, Knut Neumann, and 1 others. 2024. Following the impact chain of the la cockpit: an intervention study investigating a teacher dashboard’s effect on student learning. *Journal of Learning Analytics*, 11:215–228.
- Charles Kreidler. 1998. *Introducing English semantics*. Routledge.
- 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 the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 9662–9669.
- Rafael Ferreira Mello, Elyda Freitas, Luciano Cabral, Filipe Dwan Pereira, Luiz Rodrigues, Mladen Rakovic, Jackson Raniel, and Dragan Gašević. 2024. Words of wisdom: A journey through the realm of natural language processing for learning analytics—a systematic literature review. *Journal of Learning Analytics*, 11(3):82–105.
- W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. [Definitions, methods, and applications in interpretable machine learning](#). *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.
- Christopher Ormerod. 2022. Short-answer scoring with ensembles of pretrained language models. *ArXiv preprint arXiv:2202.11558*.
- Christopher Ormerod. 2025. [Automated essay scoring incorporating annotations from automated feedback systems](#). In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Coordinated Session Papers*, pages 9–18, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- W James Popham. 1997. What’s wrong-and what’s right-with rubrics. *Educational Leadership*, 55:72–75.
- Renjith Ravindran and Ikkyu Choi. 2025. [Investigating adversarial robustness in LLM-based AES](#). In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Coordinated Session Papers*, pages 86–91, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- Y Malini Reddy and Heidi Andrade. 2010. A review of rubric use in higher education. *Assessment & Evaluation in Higher Education*, 35(4):435–448.
- Brian Riordan, Michael Flor, and Robert Pugh. 2019. How to account for misspellings: Quantifying the benefit of character representations in neural content scoring models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 116–126, Florence, Italy. Association for Computational Linguistics.
- Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.
- Shashank Sonkar, Kangqi Ni, Lesa Tran Lu, Kristi Kincaid, John S Hutchinson, and Richard G Baraniuk. 2024. Automated long answer grading with ricechem dataset. In *International Conference on Artificial Intelligence in Education*, pages 163–176. Springer.
- N. Sreedevi, M. Madhusudhan Rao, Sridevi Dasam, Roopa Traisa, Jasgurpreet Singh Chohan, V. Saranya, and Ahmed I. Taloba. 2026. [Rubric-relational discourse modeling with counterfactual explainability for multi-trait automated essay scoring](#). *International Journal of Advanced Computer Science and Applications*, 17(2).

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568(127063):127063.

Tianqi Wang, Naoya Inoue, Hiroki Ouchi, Tomoya Mizumoto, and Kentaro Inui. 2019. Inject rubrics into short answer grading system. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 175–182, Hong Kong, China. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Yuchen Wei, Dennis Pearl, Matthew Beckman, and Rebecca J Passonneau. 2025. Concept-based rubrics improve llm formative assessment and data synthesis. *arXiv preprint arXiv:2504.03877*.

Julia Wunderle, Anton Ehrmanntraut, Jan Pfister, Fotis Jannidis, and Andreas Hotho. 2025. New encoders for german trained from scratch: Comparing modernbert with converted llm2vec models. *arXiv preprint arXiv:2505.13136*.

Fabian Zehner, Hyo Jeong Shin, Emily Kerzabi, Andrea Horbach, Sebastian Gombert, Frank Goldhammer, Torsten Zesch, and Nico Andersen. 2025. [Down the cascades of omethi: Hierarchical automatic scoring in large-scale assessments](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 660–671, Vienna, Austria. Association for Computational Linguistics.

Ramon Ziai, Niels Ott, and Detmar Meurers. 2012. [Short answer assessment: Establishing links between research strands](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 190–200, Montréal, Canada. Association for Computational Linguistics.

A Appendix

Sensitivity analysis of $k_{\max}$ and λ . To examine the sensitivity of the model to the two central structural hyperparameters, we conducted a targeted ablation for *Joint-Cross-AGRAA* on the ALICE-LP unseen-answers setting. We varied the cap on the number of retained rubric directions $k_{\max} \in \{3, 4, 5, 6\}$ and the entropy regularisation weight $\lambda \in \{0.0, 0.2, 0.6\}$ while keeping all other

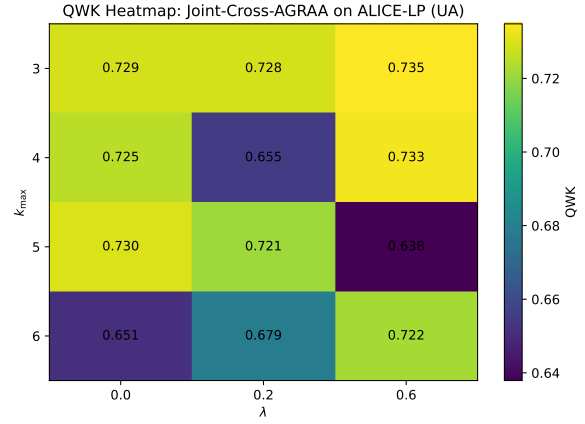


Figure 3: QWK results for the different $k_{\max}$ and λ combinations.

hyperparameters fixed. Due to computational constraints, the analysis was performed with a single random seed and should therefore be interpreted as a compact sensitivity check rather than an exhaustive hyperparameter search. Across most configurations, performance remains within a relatively narrow range, with some outliers present. This suggests that $k_{\max}$ and λ can influence performance and should therefore not be set randomly, but overall, the model is able to handle varying configurations of these hyperparameters.

Domain-Adaptive Pre-training for Automated Short Answer Grading in Conceptual Physics: Reliability, Question-Level Analysis, and Error Reduction

Shirin Deochand Lade

shirin.lade@open.ac.uk

Alistair Willis

alistair.willis@open.ac.uk

Jonathan Nylk

jonathan.nylk@open.ac.uk

Oli Howson

oli.howson@open.ac.uk

The Open University
United Kingdom

Abstract

Automated short answer grading (ASAG) has attracted increasing research attention as a potential tool to support teachers in evaluating student responses. However, the usefulness of such systems depends on how reliably they operate under realistic classroom constraints, where only limited numbers of graded responses may be available for each question. This study investigates automated grading of conceptual physics explanations from free-response questions derived from the Force Concept Inventory (FCI), a diagnostic tool for assessing understanding of mechanics concepts.

We compare a standard BERT-based grading model with a version further adapted through subject-specific pre-training on physics textbook materials. Building on prior work showing that BERT representations for ASAG can be improved through continued pre-training on domain resources, we do not propose a new adaptation method; instead, we evaluate its reliability in a classroom-derived conceptual physics setting. The models are evaluated across varying training set sizes to examine how domain adaptation affects performance when labelled data is limited. Our results show that domain-adaptive pre-training provides the largest gains in low-data conditions, improving classification accuracy and reducing grading errors. As more labelled responses become available, the difference between the models decreases. This suggests that subject-specific pre-training primarily improves data efficiency rather than the best achievable performance.

Error analysis shows that the performance gains come from real reductions in grading mistakes. When comparing both models, the improved model corrects substantially more baseline errors than it introduces new ones, with a fixed-to-broken ratio of 4.61, indicating a clear net reduction in grading errors. This suggests that the improvement primarily reflects net error reduction rather than simply shifting mistakes.

However, improvements vary across individual assessment questions, highlighting the importance of question-level evaluation when deploying automated grading systems in classroom environments. These findings indicate that, relative to a general BERT baseline, textbook-based continued pre-training can support educational assessment.

1 Introduction

Automated Short Answer Grading (ASAG) systems are increasingly explored as tools for supporting student assessment by reducing marking workload and enabling timely feedback to students (Seneviratne and Manathunga, 2025). In real educational settings, however, automated grading systems operate under constraints that differ substantially from those typically considered in benchmark-driven experimental evaluations (Bonthu et al., 2021). When new questions are introduced or when assessments are administered to small cohorts, teachers often have access to only limited numbers of previously graded responses for each assessment question (Egaña et al., 2023). Under such conditions, automated grading systems must operate reliably with minimal task-specific supervision while still producing grading decisions consistent with human evaluation (Morris et al., 2025).

Several approaches have been proposed to improve model performance under limited supervision, including data augmentation, semi-supervised learning, and transfer learning (Bonthu et al., 2021; Burrows et al., 2015). In this work, we focus on domain-adaptive pre-training (DAPT) (Beltagy et al., 2019; Lee et al., 2020) where a pre-trained language model is further trained on domain-relevant text before task-specific fine-tuning (Gururangan et al., 2020). DAPT is an established method, and its use for ASAG has direct precedent: Sung et al. (2019) showed that BERT for short

answer grading can be improved by updating the pre-trained model using domain resources such as textbooks and question-answer data before supervised fine-tuning. Our contribution is therefore not a new adaptation algorithm. Instead, we examine whether textbook-based DAPT improves grading reliability relative to a general BERT baseline in a classroom-derived conceptual physics setting, especially when only limited labelled responses are available per question. The evaluation focuses on binary grading of held-out responses to the same assessment questions, rather than partial-credit scoring or transfer to unseen questions. We further analyse whether improvements correspond to net reductions in grading errors, how gains vary across questions, and whether the adapted model provides better calibrated confidence estimates.

In educational assessment, grading performance must be evaluated not only by predictive accuracy but also by reliability, as automated systems support decisions that influence feedback and student evaluation (Holmes et al., 2021). When comparing automated grading models, it is important to determine whether improved performance reflects a net reduction in grading errors, rather than a redistribution of errors across responses. Furthermore, grading reliability may vary across assessment questions due to differences in conceptual difficulty or response distributions, making question-level analysis important for evaluating automated grading systems in practice (Zhu et al., 2022).

This study investigates automated grading of conceptual physics explanations from free-response questions based on the FCI, a widely used diagnostic instrument in physics education (Hestenes et al., 1992; Hestenes and Halloun, 1995). Many ASAG studies rely on proprietary datasets, which can limit reproducibility and comparability across studies. In this work, we instead use FCI-based conceptual free-text questions, which are widely recognised and explicitly designed to probe common misconceptions in mechanics (Parker et al., 2023; Pathak et al., 2026). This established conceptual framework enables a more meaningful analysis of how domain-adaptive language modelling interacts with disciplinary reasoning tasks.

We address the following research questions:

- RQ1. When only limited numbers of graded student responses are available for the same assessment questions, does textbook-based

DAPT improve binary ASAG reliability relative to a general BERT baseline?

- RQ2. When model performance improves, do these gains reflect consistent reductions in grading errors across questions, or do they arise from improvements in some questions offset by declines in others?

This study makes three contributions:

- We provide a reliability-focused evaluation of textbook-based DAPT for ASAG in a classroom-derived conceptual physics setting, using free-response questions derived from the Force Concept Inventory.
- We analyse how textbook-based DAPT changes grading behaviour relative to a general BERT baseline under limited labelled data by comparing learning curves, agreement metrics, calibration, and error-overlap patterns across 26 training-set sizes and 10 random seeds.
- We examine question-level variation in grading behaviour to identify whether improvements are consistent across conceptual physics questions or concentrated in particular items.

2 Related Work

2.1 Automated Short Answer Grading

Early ASAG systems relied on rule-based approaches, lexical overlap, or semantic similarity measures (Burrows et al., 2015). Subsequent work introduced supervised learning models trained on labelled student responses (Bonthu et al., 2021). More recent approaches use transformer-based models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), which achieve strong performance on benchmark datasets (Takano et al., 2022). Recent work has also examined large language models for ASAG, with mixed evidence on whether few-shot LLM prompting outperforms fine-tuned encoder models in educational grading tasks (Grévisse et al., 2024; Kakarla et al., 2025). Despite these advances, it remains unclear how improvements in model architecture translate into practical grading behaviour, particularly whether they lead to more reliable and consistent decisions on real student responses.

However, Ley et al. (2023) identify remaining concerns regarding the reliability, cross-question consistency, and robustness of ASAG systems under limited training data in real educational environ-

ments. Many evaluations rely on curated datasets rather than authentic classroom data (Padó, 2022; Ferrara and Qunbar, 2022). Several benchmark datasets have been developed for ASAG research, including the SemEval Student Response Analysis dataset (Dzikovska et al., 2013) and the ASAP short-answer dataset (Mohler et al., 2011). Automated grading systems are generally proposed as decision-support tools to assist instructors by prioritising responses for review rather than replacing human judgement entirely (Weegar and Idestam-Almquist, 2024). In this context, overall benchmark accuracy is insufficient; evaluation must also consider reliability and error behaviour, including consistency across questions and whether models reduce rather than shift errors (Jung et al., 2025). Although instructors grade student responses during routine assessments, newly introduced questions often have few-or no-labelled responses available to train automated grading systems (Camus and Filighera, 2020; Zhang et al., 2022). The challenge of performing automated short-answer grading with only a small number of labelled student responses is compounded by the lack of systematic investigation into how ASAG systems behave across varying training set sizes, especially under realistic classroom constraints.

2.2 Domain-Adaptive Pre-training (DAPT)

Domain-adaptive pre-training (DAPT) extends the training of a language model by continuing pre-training on domain-relevant corpora prior to task-specific fine-tuning (Gururangan et al., 2020). This approach allows models to better capture specialised vocabulary and conceptual structures that may not be well represented in general web-based training data. DAPT has been shown to improve performance in specialised domains such as scientific text processing (Beltagy et al., 2019) and biomedical text analysis (Gu et al., 2021).

Sung et al. (2019) showed that BERT for ASAG can be improved by further pre-training on domain resources such as textbooks and question-answer pairs. More broadly, Gururangan et al. (2020) showed that DAPT improves performance across domains and resource settings. Our work therefore does not propose a new adaptation method; instead, it evaluates textbook-based DAPT in classroom-derived conceptual physics ASAG, focusing on low-data reliability, calibration, error overlap, and question-level variation. We do not compare against stronger science-domain encoders, physics-

domain encoders, modern LLM graders, or same-size non-physics continued pre-training controls; therefore, our claims are limited to the benefit of textbook-based DAPT over a general BERT baseline, rather than superiority over alternative encoder, LLM, or corpus-control baselines.

Recently, Hellert et al. (2024) explored domain-adapted models for scientific and technical domains, including physics-related NLP tasks. Pre-training language models on educational physics materials may therefore enable models to better represent the conceptual language used in student explanations within Force Concept Inventory (FCI) (Hestenes et al., 1992) assessments (Zhu et al., 2022). However, studies such as Henkel et al. (2024) and Gao et al. (2024) focus primarily on performance, providing limited insight into how domain-adaptive pre-training affects automated grading behaviour.

2.3 Reliability, Agreement, and Question-Level Variation in ASAG

Reliability is a fundamental requirement in educational assessment and measurement theory (Nitko and Brookhart, 2014; Schneider et al., 2023). Agreement metrics such as Cohen’s κ (Cohen, 1960) are commonly used to measure agreement between automated grading systems and human raters (del Gobbo et al., 2023). However, such metrics do not explain how errors are distributed, making it important to examine the types of mistakes systems make. Because grading decisions directly affect how students are assessed, improvements in model performance must be interpreted not only in terms of overall accuracy but also in terms of the types of errors made and the degree of agreement with human grading (Li et al., 2023). Two systems may achieve similar overall accuracy while producing different error patterns, and some grading errors may have more serious consequences for students than others.

Analysing error overlap (Dietterich, 1998) between models helps determine whether improved performance reflects a genuine reduction in grading mistakes. A model may appear more accurate because it correctly grades responses that were previously misclassified. However, it may also introduce new errors on other responses. As long as more errors are corrected than introduced, overall accuracy will increase, even if errors are redistributed across responses rather than uniformly reduced, which may lead to inconsistent or unfair grading

for certain students (Grévisse et al., 2024). In addition, grading performance may vary across assessment questions due to differences in conceptual difficulty or response distributions, meaning that some questions may be graded less reliably than others, potentially leading to unfair outcomes for students. Examining automated grading behaviour at the question level therefore provides important insight into system reliability, yet such analyses remain relatively limited in ASAG research.

2.4 Conceptual Physics Assessment and the Force Concept Inventory

This study focuses on conceptual physics assessment using free-text responses derived from the FCI (Parker et al., 2023; Pathak et al., 2026). The FCI is a widely used diagnostic instrument designed to measure students' understanding of fundamental Newtonian mechanics concepts and identify common misconceptions (Hestenes and Halloun, 1995; Hestenes et al., 1992). Yasuda et al. (2023) and Parker et al. (2022) explored short-answer reformulations of the Force Concept Inventory that capture students' reasoning more directly.

Grading conceptual explanations in free text presents particular challenges because students may express the same underlying idea using different wording, or provide responses that mix correct reasoning with misconceptions (Rainey et al., 2022; Dzikovska et al., 2013). This makes it difficult for automated grading systems to distinguish between fully correct, partially correct, and incorrect answers based on surface-level features alone. As a result, conceptual physics assessment provides a suitable setting for investigating whether DAPT on physics materials can improve grading performance by better capturing domain-specific terminology and response patterns (Trewartha et al., 2022; Sung et al., 2019). Furthermore, because the FCI targets specific conceptual dimensions, it enables fine-grained analysis of how automated grading performance varies across questions targeting different mechanics concepts.

3 Methodology and Experimental Setup

This section describes the task formulation, dataset construction, DAPT procedure, model configurations, and experimental design. The experiments are designed to evaluate the effect of DAPT while keeping the training data, model architecture, and evaluation procedure consistent across conditions.

3.1 Task and Dataset

Assessment Instrument: This study investigates the use of ASAG on free-response questions from the Newtonian Mechanics Quiz (NMQ) (Pathak et al., 2026), a modified version of the widely used Force Concept Inventory (FCI) (Hestenes et al., 1992) incorporating free-response questions (Parker et al., 2022).

Both the FCI and NMQ are conceptual diagnostic instruments designed to assess students' understanding and identify common misconceptions. The free-response questions integrated into the NMQ, requiring students to construct their own response, can provide deeper insight into students' comprehension of Newtonian mechanics than the multiple-choice questions of the original FCI alone (Pathak et al., 2026), Parker et al. (2022) and Rebello and Zollman (2004).

We use 15 free-response questions from the Newtonian Mechanics Quiz (NMQ), each designed to assess a specific conceptual aspect of Newtonian mechanics. For example, one question asks students to compare the time taken for two objects of different masses to reach the ground when dropped simultaneously, targeting understanding of gravitational acceleration and the independence of mass in free fall. These questions are conceptually aligned with Force Concept Inventory (FCI)-style items in that they probe underlying physical principles rather than requiring numerical calculation.

Student Response Dataset: The dataset analysed in this work consists of student responses collected during administrations of the NMQ at 6 UK higher education institutions in the 2024/25 academic year, either in class or as an out-of-class activity. A total of 674 quiz attempts were collected from students enrolled on introductory undergraduate physics modules. The unit of analysis for modelling is a question-response instance rather than a complete quiz attempt. Since each quiz attempt can contribute responses to multiple free-response questions, the 674 quiz attempts yield multiple labelled question-response instances across the 15 questions; the fixed test set of 1,566 therefore refers to held-out question-response instances, not 1,566 separate students or quiz attempts. Data collection was conducted via an online platform.

Each student response was independently labelled as correct or incorrect by four physics experts following agreed marking guidelines. Disagreements were resolved by a panel of three re-

searchers, who determined the final label. All responses were anonymised prior to analysis to protect student privacy. An anonymised version of the dataset has been released via The Open University repository with DOI 10.21954/ou.rd.32190003 (Pathak et al., 2026).

Split protocol: For each of the 15 questions, labelled response instances were split into a training pool and a fixed test set, with approximately 80% assigned to training and 20% to testing. All scaling subsets were sampled only from the training pool, and test responses were never used during training or subset construction. The split was performed at the question-response level rather than at the student or institution level; therefore, the evaluation measures generalisation to unseen responses for the same questions, not student-disjoint, institution-disjoint, or unseen-question generalisation.

Task Formulation: We study ASAG as a binary classification task. Given a question and a student’s free-text response, the model predicts whether the response should be marked correct or incorrect. This formulation matches the available expert labels and enables controlled analysis of agreement and error behaviour, but it does not model partial-credit or rubric-based scoring.

Each instance consists of:

- the question text,
- a student-written response,
- a binary correctness label assigned by human graders.

Table 1 reports the distribution of correct and incorrect responses per question. Class distributions vary substantially across questions, reflecting realistic classroom assessment conditions in which some concepts are more difficult than others and certain misconceptions appear more frequently.

3.2 DAPT on LibreTexts Physics

Following prior work on continued pre-training for domain adaptation and ASAG, we apply textbook-based DAPT before task-specific fine-tuning (Sung et al., 2019; Gururangan et al., 2020). The goal is not to introduce a new pre-training objective, but to test whether exposure to undergraduate physics language improves grading reliability for conceptual physics responses under low-data conditions.

Pre-training Corpus: The domain-adapted model undergoes continued pre-training on six openly available physics textbooks from LibreTexts (Dourmashkin, 2020; Gea-Banacloche, 2019; Cline, 2019), covering foundational topics in New-

tonian mechanics. LibreTexts was selected because it provides openly licensed, comprehensive undergraduate-level physics material that reflects terminology and conceptual explanations aligned with FCI-style content. Continued pre-training uses the masked language modelling objective (Devlin et al., 2019). This stage is entirely unsupervised and does not involve student responses or grading labels, ensuring that any downstream performance differences are not due to additional supervised grading labels.

3.3 Model Configurations

We compare two models that share identical architecture and task-specific training procedures, differing only in whether DAPT is applied.

3.3.1 Baseline Model

The baseline model consists of a BERT encoder (Devlin et al., 2019) fine-tuned directly on the short answer grading task.

For each instance:

- The question and student response are concatenated into a single input sequence.
- The final hidden state corresponding to the [CLS] token is passed to a linear classification layer.
- Binary predictions are produced using cross-entropy loss (Cui et al., 2019).

To simulate realistic low-resource classroom conditions, we adopt a **partial fine-tuning strategy** (Howard and Ruder, 2018; Sun et al., 2019):

- The lower eight transformer layers are frozen.
- Only the top four layers and the classification head are updated.

Fine-tuning instability in small datasets has been widely documented in prior work by Dodge et al. (2020); Mosbach et al. (2021); Tinn et al. (2023).

3.3.2 Domain-Adapted Model

The domain-adapted model follows a two-stage procedure:

- Continued pre-training on the LibreTexts physics corpus using MLM (Devlin et al., 2019).
- Fine-tuning on the ASAG task using the same architecture, layer-freezing configuration, and optimization settings as the baseline model.

Holding the base architecture, fine-tuning procedure, training subsets, and evaluation protocol constant isolates textbook-based continued pre-training relative to general BERT, but not whether

Question	Correct (%)	Incorrect (%)	N	Imb. Ratio
Q1	72.3	27.7	517	0.723
Q2	52.0	48.0	244	0.520
Q3	83.9	16.1	509	0.839
Q4	70.0	30.0	496	0.700
Q5	75.2	24.8	508	0.752
Q6	83.3	16.7	497	0.833
Q7	77.0	23.0	252	0.770
Q8	40.5	59.5	487	0.405
Q9	64.2	35.8	246	0.642
Q10	84.2	15.8	493	0.842
Q11	57.5	42.5	496	0.575
Q12	96.0	4.0	496	0.960
Q13	58.4	41.6	243	0.584
Q14	91.9	8.1	258	0.919
Q15	53.9	46.1	490	0.539

Table 1: Per-question class distribution. Percentages indicate the proportion of correct and incorrect responses.

gains arise from physics-specific content rather than continued pre-training more generally.

3.4 Training Procedure

Both models are trained on the Newtonian Mechanics dataset using supervised learning. As described in Section 3.1, the dataset consists of anonymised student responses collected during classroom administrations of free-response questions derived from the FCI. Because class distributions vary across questions (see Table 1), we use class-weighted cross-entropy loss (Cui et al., 2019), with weights computed based on inverse class frequency within each training subset. This mitigates bias toward majority classes and maintains sensitivity to less frequent response types.

All hyper-parameters, including optimizer configuration, learning rate schedule, batch size, number of epochs, and early stopping criteria, are kept identical across models. Given that small training sets can lead to unstable training dynamics, including divergence during the fine-tuning (Dodge et al., 2020), we apply early stopping (Song et al., 2020) and repeat each experimental condition using multiple random seeds.

3.5 Experimental Design: Controlled Scaling Study

We run a controlled scaling study over 26 nested training-set sizes to reflect classroom data availability, from early deployment to larger cohorts. After constructing the fixed per-question test set, all remaining responses form the training pool. For each question and training size d_x , we sample up to x correct and x incorrect responses from the

training pool. Thus, d_x represents a target per-class sampling cap, not a guarantee that every question contains x examples from each class. When one class has fewer than x available responses, both classes are capped at the smaller available count. This is particularly relevant for highly imbalanced questions such as Q12, where the number of incorrect responses is small. Larger subsets are nested by adding newly sampled responses to the previous subset where available. We train both models with 10 random seeds, resulting in 26 sizes $\times$ 10 seeds $\times$ 2 models = 520 runs. The fixed test set contains 1,566 question-response instances.

The fixed test set enables paired comparisons between models. Experiment-level differences across matched training sizes and random seeds are evaluated using paired tests over run-level metrics, while fixed/broken error counts are reported descriptively because the same test responses are evaluated repeatedly across conditions. We also report correlation analyses relating training size and baseline question performance to observed DAPT gains. A summary of statistical procedures is provided in Appendix Table 5.

For each configuration, we report Accuracy, Macro F1, Balanced Accuracy, and Cohen’s κ (Cohen, 1960) as mean $\pm$ standard deviation across runs, with detailed per-question results provided in Appendix Tables 6 and 7. Increasing the number of repetitions reduces variance in performance estimates and provides a more stable assessment of model behaviour in low-data settings, where single-run results can be highly variable (Dodge et al., 2020). This controlled scaling framework enables analysis of performance differences, data efficiency, saturation effects, and variability under limited labelled data.

4 Results and Discussion

This section evaluates whether DAPT improves the practical reliability of ASAG when applied to classroom-derived student responses, and whether observed improvements reflect consistent reductions in grading errors.

4.1 Overall Performance and Reliability (RQ1)

Across all evaluation metrics, the domain-adapted model (DAPT) shows higher agreement with reference labels compared to the baseline model.

Table 2 summarises the average performance

across all runs and training sizes.

Metric	Baseline	DAPT	Δ
Accuracy	0.906	0.963	+0.057
Macro F1	0.889	0.955	+0.066
Balanced Acc.	0.895	0.954	+0.059
Cohen's κ	0.783	0.910	+0.127

Table 2: Overall average performance across all runs and training sizes.

From a grading perspective, the increase in Cohen's κ suggests improved consistency in automated decisions, which is important for maintaining fairness in assessment.

Wilcoxon Signed-Rank Test (Experiment-Level Comparison): To evaluate whether these differences are consistent across experimental conditions, we conducted a paired Wilcoxon signed-rank test (Wilcoxon, 1945) on accuracy scores, comparing matched runs and training sizes between the two models.

The paired comparison showed a consistent difference across matched conditions:

- $W = 30.5, p < 10^{-43}$

Because training subsets are nested, this result should be interpreted as evidence of a strong repeated pattern across conditions rather than as fully independent experimental replications.

Classroom interpretation: From an assessment perspective, the increase in Cohen's κ indicates stronger agreement between automated predictions and reference labels. This indicates that the domain-adapted model may provide more stable grading decisions under varying data conditions.

4.2 Effect of Training Data Size (RQ1)

Across 260 matched comparisons (10 runs $\times$ 26 training sizes, comparing the same run and dataset size across models), the domain-adapted model outperformed the baseline in the large majority of cases:

- Accuracy improved in 257/260 comparisons
- Macro F1 improved in 258/260 comparisons
- Cohen's κ improved in 258/260 comparisons

These results indicate that performance gains are observed across most experimental conditions, rather than being limited to a small subset of runs or training sizes.

Figure 1 presents learning curves across increasing training sizes. The largest performance differences occur in low-data settings:

- d_{10} : Δ Accuracy $\approx +0.28$
- d_{50} : Δ Accuracy $\approx +0.12$
- d_{470} : Δ Accuracy $\approx +0.006$

Figure 1 shows that DAPT provides the largest gains in low-data conditions, with an accuracy improvement of approximately 28 percentage points at d_{10} and 12 points at d_{50} . As training data increases, the gap narrows, reaching approximately 0.6 points at d_{470} . This pattern is supported by a strong negative correlation between training size and DAPT gain (Spearman's $\rho = -0.90$), indicating that the benefit of domain adaptation diminishes as more labelled data becomes available.

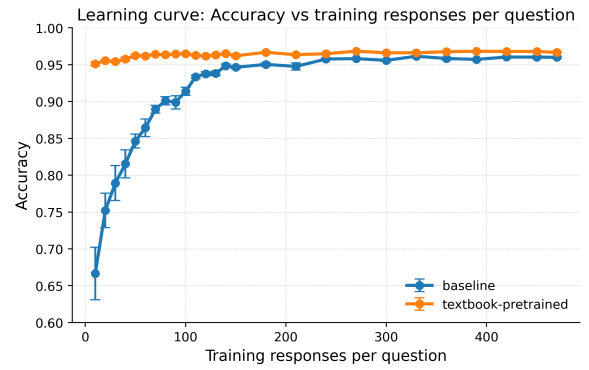


Figure 1: Learning curves across training sizes.

Classroom interpretation: From an assessment perspective, this suggests that domain-adaptive pre-training can provide more reliable grading in early deployment stages, where only a limited number of student responses are available, while offering smaller additional gains once sufficient data has been collected.

4.3 Question-Level Behaviour (RQ2)

While overall performance improves, the magnitude of improvement varies across questions. Larger gains are observed for questions where baseline performance is lower (e.g., Q4, Q5), whereas questions with already high baseline accuracy (e.g., Q1, Q14, Q15) show only marginal improvements.

A strong negative relationship between baseline accuracy and improvement (Spearman's rank correlation coefficient $\rho = -0.94$) indicates a ceiling effect, where questions that are already easy to grade offer limited room for further gains.

Figure 2 presents performance differences at the question level.

These results suggest that DAPT provides the greatest benefit for questions that are less reliably handled by the baseline. A qualitative inspection suggests that lower-gain questions may have more constrained answer spaces, whereas higher-gain questions appear to involve more varied explanations, incomplete causal reasoning, mixed correct and incorrect ideas, or domain-specific phrasing. Textbook-based DAPT may therefore help most where student responses are linguistically heterogeneous. This interpretation is exploratory, and future work should analyse response length, misconception type, and explanation structure systematically.

Classroom interpretation: From a classroom perspective, this indicates that the domain-adapted model may be particularly useful for supporting grading on more challenging questions, where automated decisions are likely to be less reliable. Importantly, we do not observe clear evidence of systematic performance degradation across questions, suggesting that improvements are not achieved at the expense of reduced performance elsewhere.

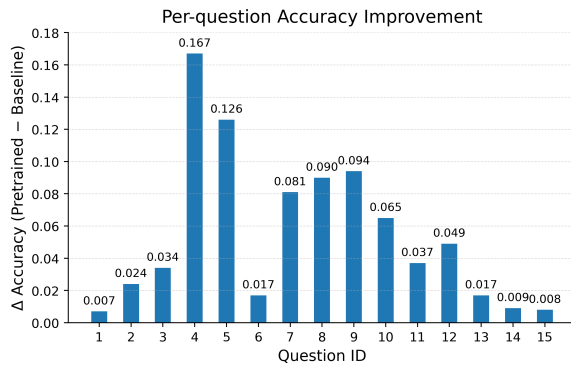


Figure 2: Question-level performance differences (δ accuracy) between baseline and domain-adapted models.

4.4 Error-Level Analysis (RQ2)

To determine whether improvements reflect genuine error reduction rather than redistribution, we conducted a global error-overlap analysis across all questions and training sizes.

In this analysis, *Fixed* refers to cases where the domain-adapted model produces a correct prediction while the baseline model is incorrect, and *Broken* refers to cases where the baseline is correct but the domain-adapted model is incorrect.

Category	Count
Both Correct	362,667
Both Wrong	8,487
Fixed	29,586
Broken	6,420
Net Reduction	23,166
Fixed/Broken Ratio	4.61

Table 3: Error overlap summary between baseline and textbook-pretrained models.

Table 3 shows that the domain-adapted model corrects substantially more baseline errors than it introduces, resulting in a net reduction of 23,166 errors and a Fixed/Broken ratio of 4.61.

Aggregate paired comparison: Because the same fixed test responses are evaluated repeatedly across training sizes and seeds, the pooled fixed/broken counts in Table 3 should be interpreted descriptively rather than as independent observations. The large imbalance between Fixed and Broken cases nevertheless shows a consistent directional pattern: across repeated paired evaluations, the domain-adapted model corrects substantially more baseline errors than it introduces.

4.5 Calibration and Confidence

Table 4 summarises calibration performance.

Metric	Baseline	DAPT
Brier score	0.0668	0.0308
ECE	0.0408	0.0228

Table 4: Calibration metrics for models used

The domain-adapted model achieves lower values on both the Brier score (Brier, 1950) and Expected Calibration Error (ECE) (Guo et al., 2017), suggesting that its predicted probabilities are better aligned with observed outcomes. In particular, the reduction in Brier score reflects improved overall probabilistic accuracy, while the lower ECE indicates that confidence estimates more closely match true correctness likelihoods.

Classroom interpretation: From a grading perspective, improved calibration is important because it allows the system’s confidence scores to be interpreted as meaningful indicators of reliability. This enables practical workflows in which low-confidence predictions can be flagged for human review, while high-confidence predictions can be trusted with greater assurance.

4.6 Discussion

These results are consistent with prior work showing that continued pre-training on domain-relevant text can improve downstream performance. [Sung et al. \(2019\)](#) demonstrated this idea for ASAG using domain resources such as textbooks, while [Gururangan et al. \(2020\)](#) showed that DAPT can improve performance across multiple domains and resource settings. The present study extends this line of work by examining reliability, calibration, error overlap, and question-level behaviour in classroom-derived conceptual physics ASAG. Because we do not evaluate LLM graders or a same-size non-physics pre-training control, the results show improvement over general BERT in this setting, but not superiority over LLM-based grading or evidence that gains are uniquely caused by physics-specific content. The evaluation also reflects same-question deployment: models are tested on held-out responses to the same questions, not on unseen questions or unseen institutions.

4.6.1 RQ1: Reliability under limited data:

The results show that domain-adaptive pretraining improves agreement with reference labels across conditions, with the largest gains observed in low-data regimes. Learning curve analysis indicates that these improvements are most pronounced when only limited labelled responses are available, and diminish as training data increases.

4.6.2 RQ2: Nature of improvements:

Improvements are observed across most questions, with larger gains for items where baseline performance is lower. Error-level analysis further indicates that these gains primarily reflect net reductions in grading errors, rather than redistribution of errors across instances.

4.6.3 Implications for practice

From a teaching perspective, these findings suggest that domain-adapted models may:

- improve consistency of automated grading decisions,
- provide the greatest benefit in early deployment scenarios with limited data, and
- support identification of responses requiring human review.

However, such systems should be used to support, rather than replace, human judgement in assessment.

5 Conclusion

We evaluated whether textbook-based DAPT improves binary ASAG reliability relative to a general BERT baseline in a classroom-derived conceptual physics setting. Across 26 training sizes and 10 random seeds, the domain-adapted model showed higher accuracy, macro F1, balanced accuracy, and Cohen's κ , with the largest gains in low-data settings.

Question-level and error-overlap analyses showed that these gains were not only aggregate improvements: the domain-adapted model corrected substantially more baseline errors than it introduced, with larger benefits for questions where baseline performance was lower. Calibration also improved, suggesting that model confidence may better support human-review workflows.

These findings support textbook-based DAPT as a useful low-data strategy for same-question ASAG deployment. However, the system should be treated as a decision-support tool, and broader evaluation is needed for stronger baselines, partial-credit scoring, unseen questions, unseen institutions, and other educational contexts.

6 Limitations

Baseline scope: We compare general BERT with a textbook-adapted version of the same model to isolate the effect of continued pre-training on undergraduate physics materials. We do not compare against science-domain or physics-domain encoders such as SciBERT or PhysBERT, stronger general encoders such as RoBERTa or DeBERTa, modern LLM-based graders, or a same-size non-physics continued pre-training control. Therefore, our results show improvement over general BERT in this setting, but do not establish superiority over stronger baselines or prove that the gains are uniquely due to physics-specific content. Future work should test these comparisons directly.

This study is conducted using free-response conceptual physics questions derived from the Force Concept Inventory (FCI), a widely used diagnostic instrument designed to assess students' conceptual understanding of Newtonian mechanics ([Hestenes et al., 1992](#); [Hestenes and Halloun, 1995](#)). This reflects a specific assessment context focused on conceptual reasoning rather than procedural or numerical problem-solving, and the findings should be interpreted within this context.

The grading task is formulated as binary clas-

sification (correct/incorrect), which enables controlled analysis of agreement and error patterns but does not capture partial-credit or rubric-based scoring. This limits direct applicability to classroom settings where teachers assign ordinal scores or apply detailed rubrics. The headline accuracy values should therefore not be directly compared with multi-class, ordinal, or rubric-based ASAG settings, which present a harder grading problem. Future work should examine whether the DAPT advantage observed here transfers to these more complex scoring settings.

Split and generalisation scope: The train/test split is performed at the question-response level. The fixed test set contains held-out responses to the same 15 questions used to construct the training pools, so the study evaluates new responses to existing questions rather than transfer to entirely unseen questions. Because the split is not student-disjoint or institution-disjoint, responses from the same student or institution may appear across training and test partitions through different question responses.

Although the experimental design includes multiple training sizes and repeated runs, the number of assessment questions is limited ($n = 15$), which constrains the strength of conclusions regarding question-level variation.

Finally, domain-adaptive pre-training is based on a single type of corpus (undergraduate physics textbooks from LibreTexts). Other domain sources may lead to different adaptation effects and remain for future investigation. The present analysis also does not establish that DAPT improves conceptual understanding in the model; gains may partly reflect better handling of physics vocabulary, common phrasings, or surface-level response patterns.

7 Ethics Statement

Collection of the Newtonian Mechanics Quiz student response dataset was approved by the relevant institutional Human Research Ethics Committee. Student responses were collected during authentic classroom assessments as part of ongoing doctoral research, and the anonymised dataset has been released via The Open University repository (Pathak et al., 2026). Participation complied with institutional ethical guidelines, and no personally identifiable information was retained. The ethical approval permits the use of anonymised data for research dissemination.

The physics textbooks used for domain-adaptive

pre-training were obtained from *LibreTexts*, an open educational resource platform providing materials under open licenses, including Creative Commons, Public Domain, and similar licenses. These licenses permit reuse, adaptation, and redistribution of content for educational and research purposes, subject to specific conditions such as attribution and, in some cases, non-commercial use or share-alike requirements. All materials were accessed and used in accordance with their respective licensing terms, and appropriate attribution was maintained where required.

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, Hong Kong, China. Association for Computational Linguistics.
- Sridevi Bonthu, Rama Sree Sripada, and M. H. M. Krishna Prasad. 2021. [Automated short answer grading using deep learning: A survey](#). In *Machine Learning and Knowledge Extraction (CD-MAKE 2021)*, volume 12861 of *Lecture Notes in Computer Science*, pages 61–78. Springer.
- Glenn W. Brier. 1950. [Verification of forecasts expressed in terms of probability](#). *Monthly Weather Review*, 78(1):1–3.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. [The eras and trends of automatic short answer grading](#). *International Journal of Artificial Intelligence in Education*, 25(1):60–117.
- Leon Camus and Anna Filighera. 2020. [Investigating transformers for automatic short answer grading](#). In *Artificial Intelligence in Education (AIED 2020)*, volume 12164 of *Lecture Notes in Computer Science*, pages 43–48. Springer.
- Douglas Cline. 2019. [Variational principles in classical mechanics](#). University of Rochester. Accessed February 2026.
- Jacob Cohen. 1960. [A coefficient of agreement for nominal scales](#). *Educational and Psychological Measurement*, 20(1):37–46.
- Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. [Class-balanced loss based on effective number of samples](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, Long Beach, California, USA. IEEE.
- Emiliano del Gobbo, Alfonso Guarino, Barbara Cafarelli, and Luca Grilli. 2023. [GradeAid: a framework for automatic short answers grading in educational contexts—design, implementation and evaluation](#). *Knowledge and Information Systems*, 65:4295–4334.
- 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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Thomas G. Dietterich. 1998. [Approximate statistical tests for comparing supervised classification learning algorithms](#). *Neural Computation*, 10(7):1895–1923.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. [Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping](#). *Preprint*, arXiv:2002.06305.
- Peter Dourmashkin. 2020. [Classical mechanics](#). Based on MIT OpenCourseWare materials. Accessed February 2026.
- Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2013. [Semeval-2013 task 7: Student response analysis](#). 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 López de Lacalle. 2023. [Exploration of annotation strategies for automatic short answer grading](#). In *Artificial Intelligence in Education (AIED 2023)*, volume 13916 of *Lecture Notes in Computer Science*, pages 377–388. Springer.
- Steve Ferrara and Ann Qunbar. 2022. [Validity arguments for AI-based automated scores](#). *Journal of Educational Measurement*, 59(3):273–294.
- Yifan Gao, Carolyn Penstein Rosé, Mingming Fan, and Kenneth R. Koedinger. 2024. [Automatic assessment of text-based responses in post-secondary education: A systematic review](#). *Computers and Education: Artificial Intelligence*, 6:100207.
- Julio Gea-Banacloche. 2019. [University physics i: Classical mechanics](#). Accessed February 2026.
- Camille Grévisse, Martina Bientzle, Juan Cendan, and Niklas Kühl. 2024. [LLM-based automatic short answer grading in undergraduate medical education](#). *BMC Medical Education*, 24:1026.
- 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 Intelligent Systems and Technology*, 12(2):1–23.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don’t stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. Association for Computational Linguistics.

- Thorsten Hellert, João Montenegro, and Andrea Pollastro. 2024. [PhysBERT: A text embedding model for physics scientific literature](#). *APL Machine Learning*, 2(4):046105.
- Markus Henkel, Chris Kedzierski, and Iryna Gurevych. 2024. [Can LLMs grade open response reading comprehension questions? an empirical study using the ROARs dataset](#). *International Journal of Artificial Intelligence in Education*.
- David Hestenes and Ibrahim Halloun. 1995. [Interpreting the force concept inventory](#). *The Physics Teacher*, 33(8):502–506.
- David Hestenes, Malcolm Wells, and Gregg Swackhamer. 1992. [Force concept inventory](#). *The Physics Teacher*, 30(3):141–158.
- Wayne Holmes, Matthew Bialik, and Charles Fadel. 2021. [Ethics of ai in education: Towards a community-wide framework](#). Technical report, UCL Knowledge Lab.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Ji Yoon Jung, Lillian Tyack, and Matthias von Davier. 2025. [Towards the implementation of automated scoring in international large-scale assessments: Scalability and quality control](#). *Computers and Education: Artificial Intelligence*, 8:100375.
- Sanjit Kakarla, Conrad Borchers, Danielle R. Thomas, Shambhavi Bhushan, and Kenneth R. Koedinger. 2025. [Comparing few-shot prompting of gpt-4 llms with bert classifiers for open-response assessment in tutor equity training](#). In *Proceedings of the Innovation and Responsibility in AI-Supported Education Workshop*, volume 273 of *Proceedings of Machine Learning Research*, pages 133–140. PMLR.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. [Biobert: a pre-trained biomedical language representation model for biomedical text mining](#). *Bioinformatics*, 36(4):1234–1240.
- Tobias Ley, Kairit Tammets, Gerti Pishtari, Pankaj Chhajra, Reet Kasepalu, Mohammad Khalil, Merike Saar, Iris Tuvi, Terje Väljataga, and Barbara Wasson. 2023. [Towards a partnership of teachers and intelligent learning technology: A systematic literature review of model-based learning analytics](#). *Journal of Computer Assisted Learning*, 39(5):1397–1417.
- Thomas W. Li, Shiyan Hsu, Max Fowler, Zhikai Zhang, Craig Zilles, and Karrie Karahalios. 2023. [Am i wrong, or is the autograder wrong? effects of AI grading mistakes on learning](#). In *Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER 2023)*, pages 159–170, New York, NY, USA. ACM.
- Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. [Learning to grade short answer questions using semantic similarity measures and dependency graph alignments](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA. Association for Computational Linguistics.
- Wesley Morris, Langdon Holmes, Joon Suh Choi, and Scott A. Crossley. 2025. [Automated scoring of constructed response items in math assessment using large language models](#). *International Journal of Artificial Intelligence in Education*, 35(2):559–586.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. [On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines](#). In *International Conference on Learning Representations*.
- Anthony J. Nitko and Susan M. Brookhart. 2014. *Educational Assessment of Students*. Pearson.
- Ulrike Padó. 2022. [Assessing the practical benefit of automated short-answer graders](#). In *Artificial Intelligence in Education (AIED 2022)*, volume 13355 of *Lecture Notes in Computer Science*, pages 212–224. Springer.
- Mark A. J. Parker, Holly Hedgeland, Nicholas Braithwaite, and Sally Jordan. 2022. [Student reaction to a modified force concept inventory: The impact of free-response questions and feedback](#). *European Journal of Science and Mathematics Education*, 10(3):310–323.
- Mark A. J. Parker, Holly Hedgeland, Sally E. Jordan, and Nicholas St. J. Braithwaite. 2023. [Establishing a physics concept inventory using computer marked free-response questions](#). *European Journal of Science and Mathematics Education*, 11(2):360–375.
- Ashutosh K. Pathak, Andrew James, Sally Jordan, and Jonathan Nylk. 2026. [Student responses to a modified force concept inventory: The newtonian mechanics quiz](#). Dataset, Version 1.
- Katherine D. Rainey, Michael Vignal, and Bethany R. Wilcox. 2022. [Validation of a coupled multiple response assessment for upper-division thermal physics](#). *Physical Review Physics Education Research*, 18(2):020116.
- N. S. Rebello and D. A. Zollman. 2004. [The effect of distracters on student performance on the force concept inventory](#). *American Journal of Physics*, 72(1):116–125.
- Johanna Schneider, Brian Schenk, and Abraham Bernstein. 2023. [Towards trustworthy autograding of short, multi-lingual, multi-type answers](#). *International Journal of Artificial Intelligence in Education*, 33(1):88–118.

- H.M.T.W. Seneviratne and S.S. Manathunga. 2025. [Artificial intelligence assisted automated short answer question scoring tool shows high correlation with human examiner markings](#). *BMC Medical Education*, 25(1146).
- Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. [How does early stopping help generalization against label noise?](#) In *Proceedings of the ICML Workshop on Uncertainty and Robustness in Deep Learning (UDL)*.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. [How to fine-tune BERT for text classification?](#) In *Chinese Computational Linguistics*, volume 11856 of *Lecture Notes in Computer Science*, pages 194–206, Cham. Springer.
- Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. 2019. [Pre-training BERT on domain resources for short answer grading](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6071–6075, Hong Kong, China. Association for Computational Linguistics.
- Yuki Takano, Tomoya Mizumoto, and Ryo Nagata. 2022. [Automatic scoring of short answers using justification cues and BERT-based models](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 12–22, Seattle, Washington, USA. 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.
- Amalie Trewartha, Nicholas Walker, Haoyan Huo, Sanghoon Lee, Kevin Cruse, John Dagdelen, Alexander Dunn, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. 2022. [Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science](#). *Patterns*, 3(4):100488.
- Rebecka Weegar and Jakob Idestam-Almqvist. 2024. [Reducing workload in short answer grading using machine learning](#). *International Journal of Artificial Intelligence in Education*.
- Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- J. Yasuda, M. M. Hull, and N. Mae. 2023. [Visualizing depth of student conceptual understanding using subquestions and alluvial diagrams](#). *Physical Review Physics Education Research*, 19(2):020121.
- Mengxue Zhang, Sami Baral, Neil Heffernan, and Andrew S. Lan. 2022. [Automatic short math answer grading via In-Context meta-learning](#). In *Proceedings of the 15th International Conference on Educational Data Mining (EDM 2022)*.
- Xinhua Zhu, Han Wu, and Lanfang Zhang. 2022. [Automatic short-answer grading via BERT-based deep neural networks](#). *IEEE Transactions on Learning Technologies*, 15(3):364–375.

A Additional Per-Question Results

Panel A (Performance Metrics). Panel A reports per-question performance as mean $\pm$ standard deviation across runs and training sizes. We report accuracy, macro F1, weighted F1, and Cohen’s κ .

Consistent with the results in Section 4, improvements in accuracy are accompanied by increases in agreement measured by Cohen’s κ , indicating that gains are not limited to raw accuracy. Variability is higher for questions with lower baseline performance, reflecting greater instability under limited data conditions, as observed in the learning-curve analysis.

Panel B (Question Characteristics and Error Dynamics). Panel B links question properties to the mechanism of performance changes. N denotes the total number of repeated paired predictions aggregated across training sizes and runs; it should not be interpreted as the number of independent test instances. N_0 and N_1 denote the repeated counts of incorrect and correct labels, respectively. Imbalance Ratio and PosRate describe class prevalence.

While ΔAcc , reported in percentage points, indicates the magnitude of improvement, it does not by itself capture grading behaviour. The Fixed/Broken decomposition provides this insight:

- **Fixed:** baseline errors corrected by the domain-adapted model
- **Broken:** new errors introduced by the domain-adapted model
- **Net:** Fixed – Broken, indicating the directional change in errors

Because the same fixed test responses are evaluated repeatedly across training sizes and random seeds, the fixed/broken counts are interpreted descriptively rather than as independent observations for significance testing. Consistent with the error-overlap analysis in Section 4, all questions show positive Net values, indicating that the domain-adapted model corrects more baseline errors than it introduces. Larger performance gains correspond to larger positive Net values, particularly for questions where baseline performance is lower. Taken together, Panels A and B support the finding that performance gains primarily reflect net reductions in grading errors across questions, rather than improvements in some questions offset by declines in others.

Table 5: Summary of statistical analyses used in the study.

Analysis	Method	Unit	n	Reported	What is assessed
Overall experiment-level comparison	Wilcoxon signed-rank	Matched run $\times$ training-size condition	260	W, p	Whether DAPT shows a repeated performance advantage over the baseline across matched experimental conditions
Global error-overlap comparison	Descriptive fixed/broken analysis	Aggregated repeated paired predictions	407,160	Fixed, Broken, Net, Fixed/Broken ratio	Whether DAPT corrects more baseline errors than it introduces; interpreted descriptively because test responses are repeated across training sizes and seeds
Per-question error behaviour	Descriptive fixed/broken analysis	Question-level aggregated repeated paired predictions	15 questions	ΔAcc , Fixed, Broken, Net	Whether improvements are concentrated in particular questions or reflect positive net error reduction across questions
Training size vs. improvement	Spearman ρ	Training-size condition	26	ρ	Monotonic association between training size and performance improvement (ΔAcc)
Baseline accuracy vs. question gain	Spearman ρ	Question	15	ρ	Monotonic association between baseline question performance and improvement under DAPT

Notes: Fixed/broken counts are interpreted descriptively because the same fixed test responses are evaluated repeatedly across training sizes and random seeds. ΔAcc is reported in percentage points. n denotes the number of analysis units, which differs by row.

Table 6: Appendix per-question results: performance metrics.

Panel A: Per-question metrics

Columns: Acc = accuracy; F1 = macro F1; wF1 = weighted F1; κ = Cohen’s kappa.

Q	Baseline BERT				Textbook-Pre-trained BERT			
	Acc	F1	wF1	κ	Acc	F1	wF1	κ
Q1	0.987 $\pm$ 0.055	0.985 $\pm$ 0.059	0.987 $\pm$ 0.063	0.974 $\pm$ 0.088	0.994 $\pm$ 0.005	0.993 $\pm$ 0.006	0.994 $\pm$ 0.005	0.986 $\pm$ 0.013
Q2	0.942 $\pm$ 0.055	0.940 $\pm$ 0.062	0.941 $\pm$ 0.062	0.883 $\pm$ 0.109	0.966 $\pm$ 0.012	0.966 $\pm$ 0.012	0.966 $\pm$ 0.012	0.933 $\pm$ 0.024
Q3	0.954 $\pm$ 0.056	0.921 $\pm$ 0.082	0.956 $\pm$ 0.051	0.845 $\pm$ 0.156	0.988 $\pm$ 0.009	0.978 $\pm$ 0.016	0.988 $\pm$ 0.009	0.956 $\pm$ 0.032
Q4	0.786 $\pm$ 0.202	0.751 $\pm$ 0.212	0.773 $\pm$ 0.225	0.556 $\pm$ 0.329	0.954 $\pm$ 0.018	0.944 $\pm$ 0.021	0.953 $\pm$ 0.017	0.887 $\pm$ 0.041
Q5	0.843 $\pm$ 0.159	0.799 $\pm$ 0.178	0.837 $\pm$ 0.173	0.633 $\pm$ 0.281	0.969 $\pm$ 0.017	0.959 $\pm$ 0.023	0.969 $\pm$ 0.017	0.917 $\pm$ 0.045
Q6	0.946 $\pm$ 0.064	0.909 $\pm$ 0.075	0.946 $\pm$ 0.063	0.823 $\pm$ 0.129	0.963 $\pm$ 0.010	0.932 $\pm$ 0.017	0.962 $\pm$ 0.010	0.865 $\pm$ 0.035
Q7	0.830 $\pm$ 0.133	0.757 $\pm$ 0.133	0.821 $\pm$ 0.147	0.544 $\pm$ 0.181	0.911 $\pm$ 0.019	0.870 $\pm$ 0.028	0.910 $\pm$ 0.019	0.740 $\pm$ 0.055
Q8	0.800 $\pm$ 0.134	0.784 $\pm$ 0.156	0.791 $\pm$ 0.152	0.589 $\pm$ 0.274	0.890 $\pm$ 0.023	0.885 $\pm$ 0.023	0.890 $\pm$ 0.023	0.771 $\pm$ 0.046
Q9	0.833 $\pm$ 0.136	0.809 $\pm$ 0.161	0.822 $\pm$ 0.158	0.645 $\pm$ 0.267	0.928 $\pm$ 0.040	0.919 $\pm$ 0.046	0.926 $\pm$ 0.042	0.839 $\pm$ 0.090
Q10	0.903 $\pm$ 0.129	0.832 $\pm$ 0.149	0.901 $\pm$ 0.134	0.684 $\pm$ 0.236	0.968 $\pm$ 0.019	0.934 $\pm$ 0.034	0.967 $\pm$ 0.018	0.869 $\pm$ 0.066
Q11	0.917 $\pm$ 0.072	0.913 $\pm$ 0.083	0.915 $\pm$ 0.079	0.828 $\pm$ 0.155	0.953 $\pm$ 0.014	0.953 $\pm$ 0.014	0.954 $\pm$ 0.014	0.905 $\pm$ 0.028
Q12	0.922 $\pm$ 0.110	0.747 $\pm$ 0.143	0.937 $\pm$ 0.079	0.518 $\pm$ 0.247	0.971 $\pm$ 0.017	0.825 $\pm$ 0.063	0.972 $\pm$ 0.013	0.651 $\pm$ 0.124
Q13	0.965 $\pm$ 0.034	0.964 $\pm$ 0.036	0.965 $\pm$ 0.036	0.928 $\pm$ 0.067	0.982 $\pm$ 0.005	0.982 $\pm$ 0.005	0.982 $\pm$ 0.004	0.963 $\pm$ 0.009
Q14	0.990 $\pm$ 0.038	0.975 $\pm$ 0.060	0.991 $\pm$ 0.030	0.953 $\pm$ 0.109	0.999 $\pm$ 0.003	0.998 $\pm$ 0.009	0.999 $\pm$ 0.003	0.997 $\pm$ 0.018
Q15	0.987 $\pm$ 0.041	0.986 $\pm$ 0.049	0.986 $\pm$ 0.050	0.974 $\pm$ 0.077	0.995 $\pm$ 0.004	0.995 $\pm$ 0.004	0.995 $\pm$ 0.004	0.990 $\pm$ 0.009

Table 7: Appendix per-question results: question characteristics and descriptive error dynamics.

Panel B: Per-question characteristics and descriptive error dynamics

Columns: N = total repeated paired predictions aggregated across training sizes and runs; N_0/N_1 = repeated counts of label 0/1; Imb. Ratio = minority/majority class ratio; PosRate = N_1/N ; $\Delta\text{Acc (pp)}$ = pre-trained – baseline; Fixed/Broken = descriptive error changes; Net = Fixed – Broken.

Q	N	N₀	N₁	Imb. Ratio	PosRate	$\Delta\text{Acc (pp)}$	Fixed	Broken	Net
Q1	33800	9360	24440	0.383	0.723	0.70	335	98	237
Q2	15860	7540	8320	0.906	0.525	2.45	479	91	388
Q3	33280	5200	28080	0.185	0.844	3.41	1357	221	1136
Q4	32500	9620	22880	0.420	0.704	16.74	6169	727	5442
Q5	33020	8060	24960	0.323	0.756	12.55	4577	432	4145
Q6	32500	5460	27040	0.202	0.832	1.71	1167	610	557
Q7	16380	3640	12740	0.286	0.778	8.07	1765	443	1322
Q8	31720	18980	12740	0.671	0.402	8.96	4230	1388	2842
Q9	16120	5720	10400	0.550	0.645	9.44	2226	704	1522
Q10	32240	4940	27300	0.181	0.847	6.50	2599	505	2094
Q11	32500	13780	18720	0.736	0.576	3.69	1680	480	1200
Q12	32500	1300	31200	0.042	0.960	4.91	2204	609	1595
Q13	15860	6500	9360	0.694	0.590	1.71	291	20	271
Q14	16900	1300	15600	0.083	0.923	0.91	160	6	154
Q15	31980	14820	17160	0.864	0.537	0.82	347	86	261

Notes: Fixed/Broken counts are aggregated over repeated paired evaluations and are interpreted descriptively rather than as independent observations.

Measuring Optimal Challenge: Trajectory-Based Difficulty Alignment in Open-Ended Language Tutoring

Ziqi Shu

Stanford University
ziqishu@stanford.edu

Shuman Wang

Stanford University
shuman@stanford.edu

Michael Hardy

Stanford University
hardym@stanford.edu

Abstract

Conversational English as a Foreign Language (EFL) tutoring relies on dynamically generated exercises rather than fixed item banks, so traditional difficulty estimation cannot verify whether a task is appropriately calibrated to a learner. We propose a framework that measures difficulty alignment directly from observable interactional behavior, classifying each exercise into one of three states (Under-Challenged, Optimally Challenged, or Over-Challenged) based on turn-level sequences of student attempts, errors, confusion, and tutor scaffolding. Using 1,566 exercises from the Teacher-Student Classroom Corpus, we validate the classification against human annotation (Cohen’s $\kappa = 0.79$ at the state level) and show that a learner’s cumulative trajectory of these states predicts success on subsequent exercises. Aggregating these predictions into a within-session capability-shift proxy, we find that sessions with higher proportions of over-challenging exercises systematically yield lower estimated shifts, while optimally challenging interactions are significantly associated with greater improvement than under-challenging ones—patterns consistent with Krashen’s Input Hypothesis.

1 Introduction

How can we determine computationally whether a tutoring exercise is appropriately calibrated to an individual learner? In conversational English as a Foreign Language (EFL) tutoring, exercises are not drawn from a fixed item bank but are selected or generated dynamically or organically in response to evolving learner in-context needs. Whether delivered by a tutor drawing on experience or by an LLM system generating exercises on the fly, we know of no computationally tractable method for accurately evaluating whether the tutoring content is at an optimal level of productive engagement for a learner, leaving us without a measure of instructional calibration.

Multiple traditions in learning theory converge on optimal productive engagement: from desirable difficulties (Bjork and Bjork, 2011) and productive failure (Kapur, 2008) to Vygotsky’s Zone of Proximal Development (Vygotsky, 1978). In second language acquisition, Krashen’s Input Hypothesis (Krashen, 1977) formalizes this as $i + 1$: optimal acquisition occurs when input slightly exceeds the learner’s current mastery level, i . The Interaction Hypothesis (Long, 1996) further predicts that scaffolding moves during task completion facilitate learning, highlighting the role of the tutor. These frameworks yield an operationalizable expectation: appropriately calibrated exercises should be observable in the learner’s interactional behavior in conversation with their tutor.

Existing computational approaches to difficulty estimation, however, are not designed to operate on dynamically generated exercises in open-ended conversational settings. Item Response Theory (Hambleton et al., 1991) estimates item difficulty and learner ability from response patterns over a calibrated item bank, supporting adaptive assessment rather than instructional calibration in dynamic tutoring. LLM-based difficulty estimation (Brant et al., 2026; Liu et al., 2025) sidesteps the item-bank requirement but infers difficulty from the model’s own capabilities rather than from learner-specific evidence. Knowledge tracing (Corbett and Anderson, 1994; Piech et al., 2015), including recent LLM-based variants (Neshaei et al., 2024), addresses a related but distinct problem: estimating a learner’s mastery of a fixed taxonomy of skills from their history of correct and incorrect responses, in order to guide which skills to practice next. It is well suited to domains where competencies decompose into discrete, repeatedly practiced units, but open-ended conversational practice offers neither a fixed skill taxonomy nor sufficient per-skill repetition, and even where it could be applied, it would tell us what a learner knows rather than whether a

given exercise was appropriately challenging at the moment of delivery. Crucially, none of these approaches leverage the rich behavioral signal available in the interaction itself—treating difficulty as either an intrinsic property of the item or an aggregate of historical response patterns, rather than as something that manifests in real time through the learner’s interactional behavior.

We argue that the interaction surrounding an exercise—the sequence of student attempts, errors, confusion signals, and tutor scaffolding moves—provides a theoretically grounded and reproducible operationalization of Krashen’s optimal challenge. The core insight is that while “difficulty” and “productive struggle” are latent constructs without direct ground truth, their constituent behaviors are concrete and observable: a student producing an incorrect response, asking what a word means, or requesting clarification are all explicit, annotatable events. By mapping these turn-level behavioral sequences to optimal challenge alignment states—*Optimally Challenged* ($i + 1$; struggle followed by success), *Under-Challenged* (i ; correct without prior struggle), and *Over-Challenged* ($i + \mathbb{Z}_{>1}$ unresolved difficulty)—we obtain a computational measure grounded directly in the observable indicators that the Input and Interaction Hypotheses identify as markers of productive learning.

However, classifying individual exercises is only the initial step; isolated difficulty states are momentary snapshots. To test the underlying learning theories, we must understand how these states accumulate to shape learning trajectories. We achieve this through a two-stage framework. At the micro-level, we examine whether a learner’s cumulative trajectory of difficulty states, combined with performance and scaffolding features, predicts success on subsequent exercises. At the macro-level, we leverage this predictive model to construct an in-situ *capability-shift proxy*, the shift in a student’s expected success probability across a session, and test whether sessions with higher proportions of optimally challenged interactions yield greater estimated improvement. Together, this framework motivates three research questions:

- **RQ1 (Annotation & Classification):** Can turn-level interactional trajectories be reliably annotated and systematically mapped to difficulty alignment states that reflect the relational difficulty of tutoring exercises?
- **RQ2 (Micro-Level Prediction):** To what ex-

tent do session-level interaction histories—including prior performance, scaffolding behavior, and difficulty alignment states—predict a learner’s probability of success on subsequent exercises?

- **RQ3 (Macro-Level Inference):** How does the overall composition of difficulty alignment states within a tutoring session relate to estimated learner improvement across the session, controlling for structural confounders?

2 Data

The dataset used in this study is the **Teacher-Student Chatroom Corpus (TSCC) Version 2** (Caines et al., 2020), which contains message-based English Language tutoring sessions. The dataset includes a total of 260 sessions, involving 2 teachers and 12 students.

Each session is stored as a TSV file, where each row represents an anonymized message sent by either a student or a tutor. The dataset provides the following attributes for each message: `user.id` (unique identifier for the sender), `role` (student/teacher), and `seq.type` (the message type, where the label *exercise* indicates that the teacher is providing an exercise in the current sequence.) Additionally, the dataset includes a metadata file, which provides annotations for each session, including: Student’s CEFR (Common European Framework of Reference for Languages, (Council of Europe, 2001)) level, age, and native language (See Table 1).

ID	CEFR	Age	L1	#Sess.	#Ex.
002	B1	22	Japanese	3	27
003	B2	32	Japanese	4	26
004	C1	40	Spanish	28	53
005	B2	20	Italian	6	34
006	B1	23	Thai	4	29
007	B2	22	Mandarin Chinese	32	297
009	B2	12	Ukrainian	48	518
010	B2	13	Ukrainian	14	138
011	B2	26	Russian, Ukrainian	51	217
012	C2	33	Russian	26	47
014	C2	30	Italian	26	72
015	B2	30	Mandarin Chinese	18	108

Table 1: Student Demographic Summary in TSCC v2. # Sess. indicates the number of sessions, and # Ex. indicates the number of exercises.

3 Methods

Our methodology proceeds in two analytical stages. The *micro-level* stage classifies each exercise into a difficulty alignment state (Sections 3.1–3.3) and fits a Generalized Linear Mixed-Effects Model that predicts next-exercise success from cumulative trajectory features (Section 3.4), establishing that these features carry predictive signal. The *macro-level* stage then aggregates the micro model’s out-of-fold predictions into a session-level capability-shift proxy (Section 3.5) and regresses it on cumulative difficulty proportions, controlling for exercise-length changes, tutor identity, and student-level variance (Section 3.6). This two-stage design isolates the pedagogically meaningful contribution of difficulty composition from co-varying surface confounds that the micro model alone cannot disentangle.

3.1 Data Preprocessing and Exercise Identification

Starting from 2,527 sequences labeled as containing exercises in the original corpus, we applied a filtering pipeline to isolate exercises that are self-contained within the chat log, contain an explicit exercise body, and require productive English practice. Following established methodologies that validate LLMs as reliable data annotators against human-labeled subsets (Gilardi et al., 2023; Wang et al., 2023; Törnberg, 2023), we employed a hybrid annotation process: one author manually judged eligibility for a 10% random sample ($n=252$), then validated an automated classifier (GPT-4.1-2025-04-14; prompt documented in Appendix C.1) against these judgments. We required precision > 0.85 on the verification set before applying the classifier at scale, prioritizing precision over recall to keep the final exercise set conservative; the threshold was met (full validation metrics reported in Section 4.1). The automated pipeline was then applied to the remaining sequences, yielding a final dataset of **1,566** tutor-led exercises. Full eligibility criteria and pipeline details are provided in Appendix A.

3.2 Turn-Level Behavioral Annotation

To capture the interactional trajectory following each exercise, we developed a turn-level coding scheme (Table 2) classifying each post-exercise turn by speaker role and pedagogical function. Annotation proceeds sequentially and terminates at

a stop condition (Student Correct Response or Teacher Direct Answer). We annotated a random sample of 157 exercises (10% of the filtered corpus).

To ensure robustness, we employed a two-stage validation process.

Human Inter-Rater Reliability Two authors independently annotated a pilot subset of 30 exercises (approximately 150 turns). Inter-rater agreement was calculated using Cohen’s κ to account for chance agreement. Full annotation proceeded only after achieving $\kappa > 0.70$ (true $\kappa = 0.96$), indicating substantial agreement. Disagreements were resolved through discussion, particularly clarifying the distinction between *Content* and *Procedure* confusion and the three types of student response.

LLM-Human Agreement To scale annotation, we evaluated GPT-5.2 (GPT-5.2-2025-12-11, non-reasoning) on the fully manually annotated set (157 exercises, 1,570 turns). The prompt is documented in Appendix C.2. The model was prompted using the same definitions and contextual constraints described above. Following the same threshold-gate criterion used for human IRR but with a stricter cutoff, we required GPT-5.2 to achieve Cohen’s $\kappa > 0.80$ (“almost perfect agreement” under the Landis and Koch scale (Landis and Koch, 1977)) against human annotations before deploying the pipeline on the full corpus; the threshold was met (full per-label results in Section 4.2).

3.3 Trajectory-to-Challenge Mapping

We map each exercise’s post-exercise turn sequence (up to $L_{max} = 10$ turns) into one of three mutually exclusive difficulty alignment states. The tripartite partition follows directly from Krashen’s Input Hypothesis (Krashen, 1977), which distinguishes input at i (under-challenging), at $i + 1$ (optimally challenging), and beyond $i + 1$ (over-challenging) relative to the learner’s current mastery i . In the human-annotated set, 93.6% of exercises reached a stop condition within this window, confirming that the threshold captures the natural resolution boundary of most interactions. The three states are:

- **Under-Challenged (i):** The student produces a correct response without prior evidence of content-level struggle (no “Response Attempt – Incorrect” or “Content Question/Confusion” in the trajectory).

Speaker	Label	Definition
Teacher	Hint/Instruction	Scaffolding, elicitation (e.g., “Tell me more”), or clarifying instructions related to the exercise.
	Direct Answer	Explicit provision of the correct answer or full solution.
Student	Response Attempt – Correct	Grammatically and semantically complete answer.
	Response Attempt – Incorrect	Attempt containing linguistic errors or incorrect content.
	Response Attempt – Partial	Correct but incomplete response (e.g., filling only one of multiple blanks).
	Content Question/Confusion	Questions or statements reflecting gaps in linguistic knowledge (e.g., “I don’t know this word”).
	Procedure Question/Confusion	Questions about task mechanics rather than language content (e.g., “Should I type it?”).
Any	NA	Turn unrelated to the specific exercise body (off-topic chat, meta-talk, references to other exercises).

Table 2: Turn-level tutor-student interaction taxonomy.

- **Optimally Challenged** ($i + 1$): The student eventually succeeds, but the preceding trajectory contains explicit markers of struggle (an incorrect attempt or content confusion).
- **Over-Challenged** ($i + \mathbb{Z}_{>1}$): The tutor provides the answer via “Direct Answer,” or the interaction reaches L_{max} without a correct student response.

Figure 1 illustrates each state with example trajectories from the corpus.

This classification operates exclusively on student-driven evidence: tutor scaffolding alone does not trigger the Optimally Challenged state, and procedural questions about task mechanics (e.g., “Should I type it?”) do not count as evidence of content-level difficulty—if a student asks such a question and subsequently solves the task, the trajectory remains Under-Challenged.

3.4 Session-Level Next-Exercise Success Prediction

Having classified each exercise into a difficulty alignment state, we next ask whether a learner’s cumulative session history, including their trajectory of the difficulty alignment state and interaction features, provides a predictive signal for success on subsequent exercises. We formalize this as a binary classification task: given the first t exercises in a session, predict whether the student will respond correctly to exercise $t + 1$.

3.4.1 Trajectory Prefix Construction and Feature Selection

For a session containing T exercises (ordered chronologically by turn number), we construct $T - 1$ prefix rows, where each row at position $t \in \{1, \dots, T - 1\}$ uses the history of exercises 1 through t to predict the outcome of exercise $t + 1$.

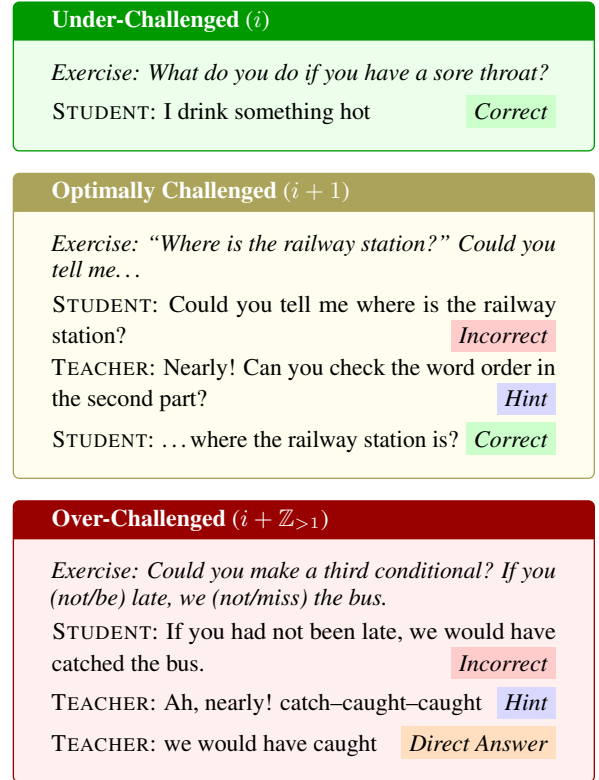


Figure 1: Illustrative trajectories for each difficulty alignment state, drawn from the TSCC corpus. Turn labels are color-coded: **Correct**, **Incorrect**, **Hint**, **Direct Answer**.

Sessions with only a single exercise are excluded. We initially extracted 24 features across four categories: historical performance, difficulty alignment trajectories, scaffolding interactions, and exercise surface properties. A comprehensive table defining all initial features is provided in Appendix D.

To prevent information leakage across related observations, we employ **session-level** 5-fold cross-validation. All prefix rows from a given session are assigned to the same fold, ensuring that no session contributes to both training and evaluation within

any fold.

To objectively identify the most robust and generalizable subset of predictors from this initial pool and to prevent overfitting, we applied L1-regularized logistic regression (Lasso) across 5-fold cross-validation. We established a strict inclusion criterion: only features selected by the Lasso penalty in at least 0.60 of the folds (i.e., ≥ 3 out of 5 folds) were retained for the final predictive model.

This data-driven selection process retained 13 features. The key retained features central to our pedagogical analysis include:

- **Cumulative Proportions of Difficulty Alignment States:** The cumulative fraction of exercises up to turn t that were classified as Over-Challenged and Under-Challenged, respectively. The proportion of Optimally-Challenged was dropped by Lasso to avoid collinearity with the other two. By definition, these three proportions represent the complete composition of the session history, summing to 1.
- **Consecutive Optimally Challenged States:** The count of uninterrupted Optimally Challenged states immediately preceding $t + 1$.
- **Consecutive Successes:** The count of consecutive correct responses immediately preceding $t + 1$.

Other retained features include scaffolding metrics (e.g., average hints per exercise), upcoming exercise character and word lengths, and the student’s baseline CEFR proficiency. A complete feature dictionary and the detailed L1 selection frequencies are provided in Appendix D.

After excluding 50 sessions containing none or single exercise, this construction yielded 1,324 prediction rows from 210 sessions. The target variable (next-exercise success) has a positive rate of 70.7%, indicating moderate class imbalance toward correct responses. Prefix lengths (the number of prior exercises available as history) range from 1 to 34 (mean ≈ 5.8), reflecting substantial variation in the amount of session context available.

3.4.2 Generalized Linear Mixed-Effects Model

We model next-exercise success using a generalized linear mixed-effects model (GLMM; [Breslow](#)

and [Clayton, 1993](#)) with a binomial response and a logit link function. Standard logistic regression assumes that all observations are independent; however, exercises are structurally dependent as they are nested within specific students and tutors. To account for this nested structure, we include a student-level random intercept to control for baseline differences in learner proficiency, and a fixed effect for teacher identity to account for unobserved pedagogical differences between the two tutors in the dataset.

Formally, for the student-random + teacher-fixed specification, let $y_{ijs} \in \{0, 1\}$ denote correctness for prediction instance i from student j with teacher s . We model

$$\text{logit}(P(y_{ijs} = 1)) = \mathbf{z}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{\alpha}_s^\top \mathbf{1}\{\text{teach.} = s\} + u_j \quad (1)$$

where $\mathbf{z}_{ij}$ contains the selected covariates (numeric covariates are standardized within each training fold), $\boldsymbol{\alpha}_s$ denotes teacher fixed effects (with a reference category), and $u_j \sim \mathcal{N}(0, \sigma_u^2)$ is a student-level random intercept.

To ensure our findings regarding key behavioral predictors are not artifacts of a specific model structure, we also tested alternative specifications, including models without random effects and models with varying control structures. These robustness checks are reported in Appendix E, demonstrating consistent feature effects across different specifications.

3.5 In-Situ Capability-Shift Proxy

Observational tutoring corpora such as TSCC typically lack formal pre- and post-assessments, precluding direct measurement of learning outcomes. We therefore construct an *in-situ capability-shift proxy*—a within-session shift in model-estimated success probability, not a measured change in language acquisition—by leveraging the out-of-fold predicted probabilities from the next-exercise success model described in Section 3.4.

The predictive model established in Section 3.4 serves as a noisy but informative indicator of a student’s real-time capability. The core intuition is that if a student’s model-estimated probability of success increases over the course of a session, the model has detected a positive shift in the student’s interactional profile toward patterns associated with correct responses—providing an operational proxy for *model-estimated* within-session capability shifts, rather than a direct measure of

language acquisition.

3.5.1 Definition

For each session trajectory, the next-exercise success model produces a sequence of out-of-fold predicted probabilities $\hat{p}_2, \hat{p}_3, \dots, \hat{p}_T$, where $\hat{p}_t$ denotes the predicted probability that the student answers exercise t correctly, conditioned on the session history through exercise $t - 1$. Note that no prediction is generated for the first exercise ($t = 1$), as it has no prior history.

We define the **capability shift** for a trajectory as the direct difference between the final predicted probability and the initial predicted probability within the session. Specifically, let $n = T - 1$ denote the number of predictions in a trajectory. We distinguish two cases:

- $n < 3$ (fewer than 4 exercises / 3 predictions): The trajectory is **discarded**, as it provides insufficient intermediate history to form a valid pedagogical composition. (57 trajectories discarded)
- $n \geq 3$: $CS = \hat{p}_T - \hat{p}_2$. (153 trajectories)

3.6 Associating Difficulty Alignment with Capability Shift

Having constructed the in-situ capability-shift proxy, we transition from micro-level, exercise-by-exercise prediction to session-level attribution. Because the micro-level model focuses on isolated exercise outcomes rather than cumulative improvement, we aggregate its predictions across full session trajectories to operationalize and observe capability shifts. This analysis evaluates how the model-estimated capability shifts are explained by the session’s overall composition—specifically the proportions of Under-Challenged, Optimally Challenged, and Over-Challenged exercises. By regressing the capability-shift proxy against these macro-level proportions, we interpret the micro-level interactional dynamics through the pedagogical framework of the Input Hypothesis.

3.6.1 Trajectory-Level Variables

For each valid session-level trajectory, we extract the following variables:

- **Difficulty alignment proportions.** The cumulative proportion of exercises classified as Under-Challenged, Optimally Challenged, and Over-Challenged across the full session.

- **Capability shift.** The in-situ capability-shift proxy as defined in Section 3.5.

- **Exercise-length deltas (control variables).**

To account for the possibility that shifts in exercise surface complexity, rather than actual capability improvements, drive changes in the predicted success probability, we compute control variables capturing how exercise length changes between the early and late windows. Using the same early/late window indices as the capability-shift computation, we calculate $\Delta_{\text{chars}} = \bar{\ell}_{\text{late}}^{\text{chars}} - \bar{\ell}_{\text{early}}^{\text{chars}}$ and $\Delta_{\text{words}} = \bar{\ell}_{\text{late}}^{\text{words}} - \bar{\ell}_{\text{early}}^{\text{words}}$, where $\bar{\ell}$ denotes the mean exercise body length (in characters or words) within the respective window.

- **Teacher identity (control variable).** It is a binary indicator (dummy variable) encoding which of the two tutors in the dataset conducted the session. This fixed effect is included to systematically control for unobserved, tutor-specific baseline differences in pedagogical style, hint generosity, or exercise formulation.

3.6.2 Controlled Linear Mixed-Effects Regression

To evaluate the relationship between session composition and estimated capability shift while accounting for the nested data structure, we regressed the capability-shift proxy on the cumulative difficulty proportions using a Linear Mixed-Effects Model (LMM). This model controls for shifts in exercise surface complexity (Δ_{chars} and Δ_{words}) and tutor identity as a fixed effect, while including a random intercept for student identity to account for multiple sessions per learner.

Because the three difficulty alignment proportions are compositionally constrained (they sum to one within each trajectory), entering all three as predictors would introduce perfect collinearity. We therefore fit the model by including two of the three proportions, omitting the third as the reference category.

Formally, omitting Under-Challenged as the ref-

erence, the model is specified as:

$$\begin{aligned}
CS_{js} = & \beta_0 + \beta_1 \cdot \text{PropOptim}_{js} \\
& + \beta_2 \cdot \text{PropOver}_{js} \\
& + \beta_3 \cdot \Delta_{\text{chars},js} \\
& + \beta_4 \cdot \Delta_{\text{words},js} \\
& + \gamma \cdot \text{Tutor}_s + u_j + \varepsilon_{js}
\end{aligned} \tag{2}$$

where j indexes the student and s indexes the tutor. Under this parameterization, β_1 estimates the expected change in capability shift when shifting proportion from Under-Challenged to Optimally Challenged exercises (holding Over-Challenged and all controls constant), and β_2 estimates the corresponding contrast from Under-Challenged to Over-Challenged.

To recover the third pairwise contrast (Optimally Challenged versus Over-Challenged), we equivalently re-parameterize the same model with Over-Challenged as the reference category. This controlled specification quantifies the theoretical variables’ impact on the expected capability shifts while mathematically isolating them from structural confounders.

4 Results

4.1 Exercise Identification

From the initial 2,527 sequences labeled as containing tutor exercises, we evaluated the filtering pipeline described in Section 3.1 on a 10% human-annotated verification set ($n = 252$), in which one author manually judged exercise eligibility by reviewing each target turn alongside a context window of 15 surrounding turns.

GPT-4.1 (prompted at temperature 0) was evaluated against these human judgments on the same verification set, achieving an accuracy of 0.84, precision of 0.88, recall of 0.85, and Cohen’s $\kappa = 0.66$. Under the Landis and Koch scale (Landis and Koch, 1977), this represents substantial agreement. The moderate κ relative to the high accuracy reflects the inherent ambiguity of boundary cases—particularly turns involving partially self-contained exercises or where the distinction between scaffolding for a prior exercise and initiation of a new one was unclear. Given the strong overall validation, especially the high precision, we applied the automated pipeline to the remaining sequences, yielding a final dataset of 1,566 eligible exercises.

Label	Supp.	Acc.	Prec.	Rec.	F1
Hint / Instruction	124	.975	.824	.871	.847
Direct Answer	26	.993	.800	.769	.784
Proc. Q. / Confusion	6	.999	.857	1.000	.923
Content Q. / Confusion	24	.993	.783	.750	.766
<i>Response Attempt:</i>					
Correct	121	.985	.930	.876	.902
Incorrect	61	.989	.891	.803	.845
Partial	23	.986	.513	.870	.645
NA	1185	.963	.992	.959	.975
Overall	1570	.932	.932	.932	.932

Table 3: LLM–human agreement on turn-level behavioral annotation ($n = 1,570$ turns from 157 exercises). Accuracy is computed in a one-vs-rest setting per label.

4.2 Turn-Level Behavioral Annotation

To scale annotation to the full corpus, we evaluated GPT-5.2 against reconciled human annotations (157 exercises, 1,570 turns). Table 3 reports per-label performance. Overall accuracy was 0.93 (Cohen’s $\kappa = 0.84$), indicating strong alignment with human judgment across the taxonomy. Performance was highest for structurally unambiguous labels such as Response Attempt – Correct (F1 = 0.90). The primary error mode was Response Attempt – Partial (F1 = 0.65), where the model frequently conflated partial responses with fully correct or incorrect ones: a distinction that requires evaluating completeness relative to the exercise prompt. Given the strong overall agreement, we applied the automated pipeline to annotate the remaining corpus.

We further validated the LLM annotations at the trajectory-state level, where they enter the downstream model. Applying the mapping rule of Section 3.3 to both human and GPT-5.2 turn labels yields a difficulty alignment state for each of the 157 human-annotated exercises. Agreement between human-derived and LLM-derived states is substantial (accuracy = 0.885, Cohen’s $\kappa = 0.786$). Per-class F1 ranges from 0.81 (Over-Challenged) to 0.92 (Under-Challenged), with Optimally Challenged at 0.85. The full state-level confusion matrix and per-class breakdown appear in Appendix B.

4.3 Difficulty Alignment State Distribution

Applying the trajectory-to-challenge mapping (Section 3.3) to all 1,566 eligible exercises yields a markedly skewed distribution: 920 exercises (58.7%) were classified as Under-Challenged, 165 (10.5%) as Optimally Challenged, and 481 (30.7%) as Over-Challenged.

The predominance of Under-Challenged exer-

cises indicates that, across the corpus, students most frequently answered correctly without exhibiting prior evidence of content-level struggle. The relatively small proportion of Optimally Challenged exercises, where students demonstrated initial difficulty but ultimately succeeded, is consistent with the theoretical expectation that the zone of productive struggle is narrow relative to tasks that are either too easy or too difficult. The substantial Over-Challenged proportion (30.7%) reflects cases where the tutor ultimately provided the answer or the interaction reached the turn limit without a correct student response.

4.4 Next-Exercise Success Prediction

The generalized linear mixed-effects model achieved a mean AUC of 0.61 and mean classification accuracy of 0.70 in the 5-fold cross-validation. The above-chance AUC confirms that the retained trajectory features carry a discriminative signal for next-exercise outcomes beyond baseline prevalence. To ensure these results are robust to model specification, we tested alternative modeling structures (detailed in Appendix E), which yielded near-identical predictive performance.

To identify which features drive the model’s predictions, we report standardized fixed-effect coefficients in Table 4. The strongest pedagogical predictor was the count of consecutive optimally challenged exercises immediately preceding the target turn ($\beta = 0.233$, $p = 0.029$): an uninterrupted sequence of productive difficulty is associated with higher expected learner success on the subsequent task. Among controls, average hints per exercise showed a positive association ($\beta = 0.201$, $p = 0.049$), consistent with the hypothesis that scaffolding facilitates subsequent independent performance. The CEFR proficiency ordinal yielded a negative coefficient ($\beta = -0.184$, $p = 0.020$), reflecting that higher-proficiency learners might face more demanding exercises in this corpus rather than indicating lower capability. Cumulative difficulty proportions were not significant predictors of immediate turn-level success ($p > 0.30$); their role in shaping macro-level session trajectories is examined in Section 3.6.

4.5 Associating Difficulty Alignment with Capability Shift

To test whether the session-level composition of difficulty explains the capability-shift proxy, we analyzed the 153 valid session-level trajectories

Fixed Effects	β	SE	p
Consecutive Optimal States	0.233	0.107	.029
Avg. Hints per Exercise	0.201	0.102	.049
CEFR Level Ordinal	-0.184	0.079	.020
Prop. Under-Challenged	0.148	0.155	.340
Prop. Over-Challenged	-0.135	0.164	.411
Next Exercise # Words	-0.111	0.183	.543
Consecutive Successes	0.110	0.096	.250
Total Teacher Hints	-0.103	0.097	.288
Next Exercise # Chars	-0.075	0.184	.683
Consecutive Failures	-0.055	0.094	.555
Total Procedural Qs	-0.038	0.070	.583
Last 3 Success Rate	0.031	0.187	.870
Total Content Qs	-0.003	0.071	.961
Intercept	0.623	0.158	< .001
Tutor Fixed Effect	0.439	0.208	.035

Table 4: Standardized fixed-effect coefficients from the GLMM predicting next-exercise success. All continuous predictors are z -scored per fold. Student random intercept variance: 0.003.

($n \geq 4$ exercises) using the controlled LMM described in Section 3.6. To evaluate all pairwise contrasts among difficulty states, the model was parameterized twice, rotating the reference category. Complete estimates are presented in Table 5.

The pairwise contrasts show two main findings:

The Detrimental Effect of Over-Challenge.

The data strongly indicates that excessive difficulty hinders capability improvement. When specifying Over-Challenged as the baseline reference, shifting the session composition toward either Under-Challenged ($\beta = 0.136$, $p < 0.001$) or Optimally Challenged ($\beta = 0.219$, $p < 0.001$) yields a substantial and highly significant increase in the capability-shift proxy.

The Superiority of Optimal Challenge. The critical theoretical test lies in the contrast between Optimally Challenged and Under-Challenged states. When holding Over-Challenged constant and utilizing Under-Challenged as the baseline, increasing the proportion of Optimally Challenged exercises is associated with a significant positive increase in the expected capability shift ($\beta = 0.083$, $p = 0.041$). This pattern is consistent with sustained productive difficulty ($i + 1$) producing greater capability improvement than simple independent mastery (i), supporting the trajectory-based difficulty alignment classification.

5 Discussion

Our results provide evidence consistent with the Input Hypothesis by establishing the hierarchy: Optimally Challenged > Under-Challenged > Over-

Table 5: Controlled LMM predicting the capability-shift proxy from difficulty alignment proportions ($n = 153$ trajectories). Panels (a) and (b) are re-parameterizations of the same model, differing only in the omitted reference category to display all pairwise contrasts. The model includes a student-level random intercept.

(a) Reference Group: Under-Challenged			
Predictor	β	SE	p
Intercept	-0.020	0.023	.386
Prop. Optimal $i + 1$	0.083	0.041	.041
Prop. Over $i + \mathbb{Z}_{>1}$	-0.136	0.032	< .001
<i>Structural Controls</i>			
Δ Characters (Δ_{chars})	-0.0005	0.0006	.378
Δ Words (Δ_{words})	0.0004	0.0031	.888
Tutor Fixed Effect	0.056	0.019	.002

(b) Reference Group: Over-Challenged			
Predictor	β	SE	p
Intercept	-0.155	0.026	< .001
Prop. Optimal $i + 1$	0.219	0.041	< .001
Prop. Under i	0.136	0.032	< .001
<i>Structural Controls</i>			
Δ Characters (Δ_{chars})	-0.0005	0.0006	.378
Δ Words (Δ_{words})	0.0004	0.0031	.888
Tutor Fixed Effect	0.056	0.019	.002

Challenged. Specifically, the robust negative association for the Over-Challenged proportion suggests excessive difficulty hinders development, while the positive contrast between Optimally and Under-Challenged states indicates that sessions with greater productive struggle are associated with greater capability shifts ($p = 0.041$). Crucially, these associations remain robust after controlling for student baseline variance, tutor identity, and exercise surface complexity. The near-zero coefficients for Δ_{chars} and Δ_{words} confirm the classification captures substantive interactional demand rather than superficial prompt changes. Together, these findings provide quantitative support for the value of optimal challenge in dynamic tutoring.

Beyond these findings, this framework provides a highly scalable approach for educational measurement. Instead of requiring the LLM to directly infer the abstract, latent difficulty of a task, the framework assigns the model a much simpler objective: classifying explicit, observable interactional behaviors. Because this specific classification task is straightforward for the LLM, the resulting turn-level annotations are reliable and amenable to automation at scale. Consequently, this automatic behavior-based classification framework is in principle transferable to other flexible, open-ended, and dynamically generated on-the-fly educational contexts.

This reliable, real-time annotation enables two direct system applications. First, the trajectory-to-challenge mapping can serve as a continuous evaluation metric for both human-led and LLM-generated exercises, providing computational feedback on whether the current pedagogical strategy maintains the learner’s optimal challenge zone. Second, these alignment states can function as a theoretically grounded reward signal for reinforcement learning frameworks; rather than simply rewarding correct answers, LLMs can be optimized to generate interaction trajectories that sustain productive struggle. Crucially, because the classification operates on the immediate post-exercise interaction rather than extensive prior history, deployed systems can utilize this framework from the first exercise of a session without facing a cold-start problem.

However, these applications must be implemented with strict caution regarding their current empirical validation. The associations identified in this study are observational, and we do not demonstrate causal effects of optimal challenge on learning outcomes, nor can we guarantee generalizability across different demographics or learning domains. Translating these within-session associations based on a model-estimated proxy into generalized claims about learning, or deploying them as definitive training targets in live systems, requires future experimental manipulation and external outcome measurement.

Limitations

Aggregated Proportions versus Sequential Dynamics. Our macro-level model aggregates each session’s difficulty history into cumulative proportions, discarding the chronological ordering of states. Yet sequence likely matters: an Over-Challenged opening followed by Optimally Challenged exercises may yield different gains than the reverse, even at identical proportions. Future work should apply sequence-aware models to capture how specific state transitions shape learning trajectories.

Annotation Visibility. Our classification rests on observable interaction in the chat transcript: typed turns, errors, and explicit questions. The transcript necessarily misses cognitive effort that does not surface in writing. A student who silently translates a word, consults an external resource, or works through a difficult construction in their head before

typing a correct answer will be classified as Under-Challenged on the visible evidence, even though the exercise may have exceeded the student’s independent capability. This systematically biases the classification toward Under-Challenged for learners who externalize less of their reasoning, and would correspondingly understate the proportions of Optimally Challenged and (where external help is used) Over-Challenged exercises. Richer multimodal signals, such as typing latencies, screen capture, and eye tracking, could close this visibility gap.

Proxy Validity. Because the TSCC lacks pre- and post-assessments, our dependent variable in RQ3 is an in-situ proxy derived from the micro-level predictive model’s out-of-fold probabilities. This creates an inferential dependency: the sensitivity of RQ3’s findings is bounded by the predictive model’s discriminative capacity (AUC = 0.61). The proxy captures short-term behavioral shifts within tightly bounded sessions (mean $\approx$ 5.8 prior exercises) rather than durable language acquisition. RQ3’s results should therefore be interpreted as associations detected through the lens of this model, not as independent evidence of learning outcomes.

Unmeasured Confounders. Our analyses are observational, and the reported coefficients identify associations rather than causal effects. The student-level random intercept absorbs stable between-learner differences and the tutor fixed effect absorbs stable between-tutor differences in baseline, but several plausible confounders remain incompletely controlled. First, within-learner ability can fluctuate across sessions in ways the random intercept does not capture. Second, tutor adaptation is dynamic. A tutor may select harder exercises for students who appear to be progressing well, which would induce reverse causation between session composition and the capability-shift proxy. Third, task type (e.g., grammar drills vs. open-ended writing prompts) likely affects both how difficulty manifests in the trajectory and how the proxy responds, beyond what exercise-length controls absorb. Establishing causal effects would require experimental manipulation of session composition or substantially richer logs of tutor decision-making.

Data Scope. The analysis relies on 12 students and 2 teachers from the TSCC Version 2. This limited sample constrains the generalizability of both the predictive model and the session-level infer-

ences. In particular, the student random intercepts are estimated from few sessions per learner, and the tutor fixed effect captures only a single pairwise contrast. Replication on larger, longitudinal corpora with standardized external assessments is necessary to establish whether these associations generalize across populations and translate into lasting acquisition gains.

Acknowledgments

This work was supported in part by the Stanford Education Data Science MS Fund.

References

- Elizabeth L. Bjork and Robert A. Bjork. 2011. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. In Morton Ann Gernsbacher, Richard W. Pew, Leaetta M. Hough, and James R. Pomerantz, editors, *Psychology and the Real World: Essays Illustrating Fundamental Contributions to Society*, volume 2, pages 56–64. Worth Publishers.
- Thiago Brant, Julien Kühn, and Jun Pang. 2026. [Estimating exam item difficulty with llms: A benchmark on brazil’s enem corpus](#). *Preprint*, arXiv:2602.06631.
- Norman E. Breslow and David G. Clayton. 1993. [Approximate inference in generalized linear mixed models](#). *Journal of the American Statistical Association*, 88(421):9–25.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. [The teacher-student chat-room corpus](#). *Preprint*, arXiv:2011.07109.
- Albert T. Corbett and John R. Anderson. 1994. [Knowledge tracing: Modeling the acquisition of procedural knowledge](#). *User Modeling and User-Adapted Interaction*, 4(4):253–278.
- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Common European Framework of Reference for Languages: Learning, Teaching, Assessment. Cambridge University Press.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. [Chatgpt outperforms crowd workers for text-annotation tasks](#). *Proceedings of the National Academy of Sciences*, 120.
- Ronald K. Hambleton, Hariharan Swaminathan, and H. Jane Rogers. 1991. *Fundamentals of Item Response Theory*. Sage Publications, Newbury Park, CA.
- Manu Kapur. 2008. [Productive failure](#). *Cognition and Instruction*, 26(3):379–424.

- Stephen Krashen. 1977. Some issues relating to the monitor model. In H. Douglas Brown, Carlos Alfredo Yorrio, and Ruth H. Crymes, editors, *Teaching and Learning English as a Second Language: Trends in Research and Practice: On TESOL '77: Selected Papers from the Eleventh Annual Convention of Teachers of English to Speakers of Other Languages, Miami, Florida, April 26–May 1, 1977*, pages 144–158. Teachers of English to Speakers of Other Languages, Washington, DC.
- J. Richard Landis and Gary G. Koch. 1977. [The measurement of observer agreement for categorical data](#). *Biometrics*, 33(1):159.
- Yunting Liu, Shreya Bhandari, and Zachary A. Pardos. 2025. [Leveraging llm respondents for item evaluation: A psychometric analysis](#). *British Journal of Educational Technology*, 56(3):1028–1052.
- Michael H. Long. 1996. The role of the linguistic environment in second language acquisition. In William C. Ritchie and Tej K. Bhatia, editors, *Handbook of Second Language Acquisition*, pages 413–468. Academic Press, San Diego.
- Seyed Parsa Neshaei, Richard Lee Davis, Adam Hazimeh, Bojan Lazarevski, Pierre Dillenbourg, and Tanja Käser. 2024. [Towards modeling learner performance with large language models](#). *Preprint*, arXiv:2403.14661.
- Chris Piech, Jonathan Spencer, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. [Deep knowledge tracing](#). *Preprint*, arXiv:1506.05908.
- Petter Törnberg. 2023. [Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning](#). *Preprint*, arXiv:2304.06588.
- Lev S. Vygotsky. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA.
- Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. [Is ChatGPT a good NLG evaluator? a preliminary study](#). In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

A Exercise Identification Details

We defined an eligible exercise as a tutor turn presenting a language practice opportunity satisfying three conditions:

1. **Content Presence:** The turn contains the actual exercise body (e.g., a sentence to complete), rather than merely announcing an upcoming task.
2. **Contextual Completeness:** The exercise is self-contained within the chat history, excluding tasks requiring external resources (textbooks, URLs, images, audio).
3. **Productive Intent:** The task requires the student to produce English (structured drills, translation, open-ended prompts), excluding scaffolding for prior exercises, social exchanges, and administrative setup.

The annotation pipeline proceeded in four stages: (1) one author manually annotated a random 10% sample ($n = 252$), reviewing each target turn with a context window of 15 surrounding turns; (2) GPT-4.1 (temperature 0) was evaluated on this verification set against the precision threshold defined in Section 3.1; (3) the automated pipeline was applied to the remaining sequences; (4) student-initiated exercises were removed to restrict the analysis to tutor-led pedagogical strategies. The prompt is documented in Appendix C.1.

B State-Level Agreement on the Verification Set

To complement the turn-label agreement reported in Section 4.2, we applied the trajectory-to-challenge mapping rule (Section 3.3) separately to the human and GPT-5.2 turn-label sequences for the 157 manually annotated exercises, obtaining a derived difficulty alignment state from each annotation source. Table 6 reports the resulting confusion matrix.

Overall agreement is substantial (accuracy = 0.885, Cohen’s $\kappa = 0.786$ under the Landis and Koch scale (Landis and Koch, 1977)). The Optimally Challenged class has high precision (0.944) and modest recall (0.773): when the model labels an exercise Optimally Challenged it almost always agrees with the human annotator (17 of 18 cases), but it misses 5 of 22 human-labeled Optimally Challenged exercises, assimilating them into the neighboring Under- or Over-Challenged categories. The model never labels a human-Under exercise as Optimally Challenged, supporting the interpretation that LLM-derived Optimal-state counts are conservative rather than inflated. The dominant disagreement pattern is between Under-Challenged and Over-Challenged (12 of 18 total errors).

Human	LLM (GPT-5.2)			Total
	Under	Optimal	Over	
Under-Challenged	90	0	9	99
Optimally Challenged	3	17	2	22
Over-Challenged	3	1	32	36
Total	96	18	43	157

Table 6: Confusion matrix between human-derived and GPT-5.2-derived difficulty alignment states on the 157-exercise verification set. Rows: human; columns: LLM. Overall accuracy = 0.885, Cohen’s $\kappa = 0.786$.

C Prompts

This appendix lists the specific prompts provided to the Large Language Model for the automated annotation pipelines described in Sections 3.1 and 3.2.

C.1 Exercise Identification Prompt

System Prompt: Exercise Identification

You are an expert annotator for educational dialogues in English language learning contexts.

Your task is to determine if the "Current Turn" is an **eligible exercise turn** posed by the {role}.

Definition of Eligible Exercise

An eligible exercise is a turn where the {role} presents a language practice opportunity that:

1. **Contains the actual exercise content** (not just an announcement or instruction or scaffolding about an upcoming or previous stated exercise)
2. **Is self-contained within the chat** (does not require external resources like textbooks, videos, links, or materials not present in the conversation)
3. **Requires the student to produce or practice English**

Types of Eligible Exercises

- **Structured exercises**: fill-in-the-blank, grammar drills, vocabulary exercises, sentence completion, translation tasks, writing prompts (e.g., "Write a paragraph about...")
- **Eliciting questions for language practice**: open-ended questions designed to get the student to practice expressing themselves in English (e.g., "Can you describe a typical day for you?", "Tell me about your favorite hobby", "What would you do if...?")

What is NOT an Eligible Exercise

1. Announcements or setup without exercise body

Turns that merely introduce an exercise without containing the exercise itself. However, if a turn combines the announcement AND the exercise body, it IS eligible.

Example sequence:

- Turn X -- Teacher: "Let's try to add the word 'yesterday' to the following sentence!" -> **NOT eligible** (announcement only)
- Turn X+1 -- Teacher: "She went back home." -> **ELIGIBLE** (this is the exercise body)
- Turn X+2 -- Teacher: "Insert the word 'yesterday' in the sentence I just sent you ." -> **NOT eligible** (instruction or scaffolding referencing previous exercise)

Contrast with:

- Teacher: "Let's try to add the word 'yesterday' to the following sentence: 'She went back home.'" -> **ELIGIBLE** (contains the exercise body in one turn)

Note: If you cannot find the exercise body in the current turn, previous context, OR post context, then the turn is NOT eligible -- this implies it requires external materials not present in the chat.

2. Scaffolding or hints for a previous exercise

Follow-up prompts that support an already-given exercise but do not constitute a new exercise themselves (e.g., "Please answer in a full sentence.", "Any other ideas?", "Can you try again?", "Think about what tense to use.")

3. Social/administrative exchanges

Casual greetings, check-ins, or procedural questions that do not involve language practice (e.g., "How are you?", "Are you ready to start?", "Can we do some exercises?", "Did you finish your homework?")

4. Exercises requiring external materials

If completing the exercise requires access to resources not fully present in the conversation:

- Textbooks, worksheets, images, charts, videos, audio files
- External links or apps (e.g., Skype messages, Instagram, websites)
- Reading passages or materials not included in the chat

Key indicators that an exercise requires external materials:

- Mentions of "chart," "image," "picture," "textbook," "video," "audio," "page," or "link"

```

- References to specific question numbers (e.g., "What do you think about question 43?") without providing the question content -- this implies they are looking at an external source
- Instructions like "Look at the reading passage" or "Based on the video we watched" when that content is not in the chat

### 5. Simple yes/no or factual questions
Questions that can be answered with a single word or simple fact and do not promote meaningful language practice (e.g., "Is this correct?", "Do you understand?", "What's 2+2?")

## Decision Process

1. **Read the Current Turn carefully.** Does it contain actual exercise content that requires the student to produce English?

2. **Check the Post Context.** If the exercise body appears AFTER the current turn, then the current turn is just an announcement -> NOT eligible.

3. **Check for external material dependencies.** Using the Previous and Post Context, determine whether the exercise references materials or content not present in the conversation -> NOT eligible.

4. **Distinguish from scaffolding.** Is this turn providing hints or follow-up for an exercise that was already given in a previous turn? -> NOT eligible.

---

Previous Context:
{prev_context}

Current Turn ({role}):
{current_turn}

Post Context:
{post_context}

---

First, write a brief reasoning (maximum 50 words) explaining your decision.
Then provide the final label.

Output format:
<analysis>
Your reasoning here (<= 50 words).
</analysis>
<judgement>
YES or NO
</judgement>

```

C.2 Turn-Level Behavioral Annotation Prompt

System Prompt: Post-Exercise Discussion Annotation

```

You are an expert linguistic annotator analyzing English as a Foreign Language (EFL) tutoring dialogues.
Your task is to analyze the "Post-Exercise Discussion" and label the pedagogical function of each turn relative to a specific "Exercise Body."

### INPUT DATA
1. Previous Context: The conversation history leading up to the exercise.
2. Exercise Body: The specific problem, sentence, or grammar task the student must solve.
3. Post-Turn Text: A sequence of up to 10 dialogue turns immediately following the exercise introduction.

### LABEL TAXONOMY
Choose exactly one label for each turn from this list:

```

```

TEACHER LABELS
- Hint/Instruction
- Direct Answer [STOP CONDITION]

STUDENT LABELS
- Procedure Question or Confusion
- Content Question or Confusion
- Response Attempt - Correct [STOP CONDITION]
- Response Attempt - Incorrect
- Response Attempt - Partial

OTHER
- NA

### DEFINITION OF LABELS
- Hint/Instruction: Scaffolding, hints, elicited questions (e.g., "Tell me more"),
  or instructions.
  Also, if the Exercise Body is vague (e.g., "Make a sentence") and the teacher
  provides the specific exercise text in the chat, label that turn as `Hint/
  Instruction`.
- Direct Answer: The teacher explicitly provides the correct answer.
- Procedure Question or Confusion: Student asks how to do the task or expresses
  confusion about the activity steps (not language knowledge).
- Content Question or Confusion: Student asks about language knowledge/content, ways
  to improve the answers, or expresses lack of knowledge specific to this
  exercise.
- Response Attempt - Correct: Student provides a grammatically and semantically
  complete answer that satisfies the exercise.
- Response Attempt - Incorrect: Student attempts to answer, but the content is wrong
  or contains typos.
- Response Attempt - Partial: Student provides a correct but incomplete answer (e.g
  ., filling only 1 of 2 blanks).
- NA: The turn cannot be labeled with any of the above labels. Turn does not fit
  above categories OR is off-topic / meta-talk.

### CRITICAL ANNOTATION RULES
1. Strict Scope: Labels must strictly relate to the Current Exercise Body.
2. Stop Condition: Process turns sequentially. STOP labeling immediately after
  assigning Direct Answer or Response Attempt - Correct.
3. Always follow the teacher's judgement on correctness of the student's answer.

### DIALOGUE INPUT
Previous Context:
{prev_context}

Exercise Body:
{exercise_body}

Post-Turns:
{turns_text}

### OUTPUT FORMAT
Return pure JSON only. No markdown and no extra keys.
{{
  "annotations": [
    {{
      "turn_id": 1,
      "speaker": "Student or Teacher",
      "text": "turn text",
      "reasoning": "brief reason (<=30 words)",
      "label": "one label from taxonomy"
    }}
  ]
}}

```

D Feature Dictionary and L1 Selection Frequencies

Table 7 reports the full pool of 24 candidate features and their L1 selection frequencies across 5-fold session-level cross-validation, using the inclusion criterion described in Section 3.4. The 13 features retained for the final GLMM are highlighted in bold. Structural control variables (the student random intercept and the tutor fixed effect) were not subject to L1 penalization and were unconditionally retained.

Feature Name	Description	Selection Ratio
<i>Learner Background</i>		
CEFR Level Ordinal	Ordinal encoding of CEFR proficiency (1=B1, 2=B2, 3=C1, 4=C2).	1.00
CEFR Level (Categorical)	Categorical representation of the student’s CEFR proficiency.	0.30
<i>Trajectory Length</i>		
Prefix Length	Number of prior exercises observed in the current session.	0.00
Number of Previous Exercises	Redundant feature mathematically identical to Prefix Length.	0.00
<i>Historical Performance</i>		
Consecutive Successes	Count of consecutive correct responses immediately preceding $t + 1$.	1.00
Consecutive Failures	Count of consecutive incorrect responses immediately preceding $t + 1$.	0.80
Last 3 Success Rate	The accuracy rate over the most recent 3 exercises.	0.60
Last 2 Success Rate	The accuracy rate over the most recent 2 exercises.	0.40
Cumulative Success Rate	The overall accuracy rate across all prior exercises in the session.	0.00
Previous Exercise Correct	Binary indicator of whether the immediately preceding exercise was correct.	0.00
<i>Historical Difficulty Alignment ($i+1$)</i>		
Prop. Over-Challenged	Cumulative proportion of Over-Challenged exercises in the trajectory.	1.00
Prop. Under-Challenged	Cumulative proportion of Under-Challenged exercises in the trajectory.	0.60
Prop. Optimally Challenged	Cumulative proportion of Optimally Challenged exercises (dropped to avoid collinearity).	0.20
Consecutive Optimal States	Count of uninterrupted Optimally Challenged states immediately preceding $t + 1$.	1.00
Consecutive Over-Challenged	Count of uninterrupted Over-Challenged states immediately preceding $t + 1$.	0.00
Previous Difficulty State	Categorical variable for the difficulty state of the immediately preceding exercise.	0.00
<i>Scaffolding & Interaction</i>		
Avg. Hints per Exercise	Average number of tutor hints provided per exercise so far.	1.00
Total Teacher Hints	Cumulative count of all tutor hints provided in the trajectory.	1.00
Total Procedural Qs	Cumulative count of procedural questions asked by the student.	0.80
Total Content Qs	Cumulative count of content questions asked by the student.	0.60
Previous Ended Direct Answer	Binary indicator of whether the previous exercise ended in direct answer from teacher.	0.40
Previous Required Hint	Binary indicator of whether the previous exercise involved any tutor hints.	0.00
<i>Next Exercise Properties (Complexity Controls)</i>		
Next Exercise # Chars	Character length of the upcoming exercise prompt.	0.60
Next Exercise # Words	Word count of the upcoming exercise prompt.	0.60

Table 7: Dictionary of all initial trajectory features and their L1 selection frequencies across 5-fold cross-validation. Features selected in ≥ 0.60 of folds (highlighted in bold) were retained for the final GLMM.

E Robustness Checks for the Next-Exercise Success GLMM

To ensure that the predictive relationships identified in Section 3.4 are not artifacts of a specific statistical structure, we evaluated four variations of the Generalized Linear Mixed-Effects Model (GLMM).

The models tested include:

1. **Student Random + Tutor Fixed (Main Model):** Includes both a student-level random intercept and a categorical tutor fixed effect.
2. **Tutor Fixed Only:** Includes only the tutor fixed effect, omitting student-level variance control.
3. **Student Random Only:** Includes only the student-level random intercept, omitting the tutor fixed effect.
4. **Base Model:** A standard logistic regression omitting both student and tutor controls.

Table 8 demonstrates that all four model specifications yielded near-identical out-of-fold predictive performance. Table 9 reports the standardized coefficient estimates (β) and p -values across the three alternative specifications. Consistent with the main findings, the count of consecutive optimally challenged states (consec_aligned_t) remains a significant positive predictor across all model variations, confirming the robustness of this core pedagogical feature regardless of the structural controls applied.

Model Specification	Mean AUC	Mean Acc.
Student Random + Tutor Fixed (Main)	0.610	0.701
Tutor Fixed Only	0.610	0.701
Student Random Only	0.599	0.703
Base Model (No Controls)	0.599	0.703

Table 8: Cross-validated predictive performance across model structures ($N = 5$ folds). Hierarchical controls do not fundamentally alter discriminative power.

Fixed Effects	Tutor Fixed Only		Student Random Only		Base Model	
	β	p	β	p	β	p
Consec. Aligned States	.234	.029	.232	.029	.233	.029
Avg. Hints per Exercise	.203	.047	.156	.112	.156	.113
CEFR Level Ordinal	-.183	.019	-.125	.094	-.125	.091
Prop. Under-Challenged	.150	.337	.173	.261	.176	.254
Prop. Over-Challenged	-.135	.412	-.098	.545	-.096	.555
Next Ex. Length: Words	-.113	.547	-.122	.495	-.121	.508
Consec. Successes	.111	.249	.133	.164	.134	.161
Total Teacher Hints	-.104	.282	-.129	.178	-.131	.170
Next Ex. Length: Chars	-.073	.702	-.152	.393	-.155	.394
Consec. Failures	-.056	.551	-.053	.573	-.053	.572
Total Procedural Qs	-.038	.582	-.053	.443	-.053	.444
Last 3 Success Rate	.029	.879	.032	.865	.031	.870
Total Content Qs	-.004	.957	.006	.933	.006	.927
Intercept	.625	< .001	.922	< .001	.924	< .001
Tutor Fixed Effect	.432	.035	—	—	—	—

Table 9: Standardized coefficients and p -values across three alternative model specifications. All continuous predictors are z -scored per fold. Significant effects ($p < .05$) in bold. The effect of consecutive aligned states remains stable across all variations. Models include student random intercepts.

PEERMATHDIAL: A Middle School Dialogue Dataset for Student Collaborative Math Problem Solving

Murong Yue¹, Desmond Alexander Mcglone², Emily Slutz², Wenhan Lyu³,
Yixuan Zhang³, Jennifer Suh², Ziyu Yao¹

¹Department of Computer Science, George Mason University

²College of Education & Human Development, George Mason University

³Department of Computer Science, William & Mary

Project website (dataset and source code): [MathVC-NSF.github.io](https://github.com/MathVC-NSF)

Abstract

Collaborative Problem Solving (CPS) is a core skill in education, where the process of peer interaction is highly important. However, existing educational dialogue datasets mostly focus on classroom instruction or tutoring (i.e., teacher/tutor-student interaction), yet datasets centering small-group, student-student interaction are limited. This thus leaves research with limited resources for studying how students interact, coordinate, and solve problems together in real educational settings. To address this, we introduce PEERMATHDIAL, the first dataset of peer CPS dialogues collected from authentic middle school math classrooms. It contains 55 dialogues from 27 students, totaling 6,406 turns. To facilitate research on CPS discourse analysis, we further build a corpus-grounded dialogue act taxonomy assisted by LLMs. Using the dataset and the dialogue act taxonomy, we demonstrate the practical applications of PEERMATHDIAL across three use cases. First, we track how dialogues evolve over time and measure the impact of teacher interventions. Second, we align dialogue actions with student surveys to reveal the connection between students' traits (e.g., confidence, leadership) and their actual behaviors. Third, by evaluating LLMs on dialogue act prediction, we glimpse at the potential of LLMs for student simulation in educational applications. Our dataset and source code will be released to the community.

1 Introduction

Collaborative Problem Solving (CPS) is recognized as a core competency in K-12 education, as students share information, coordinate participation, explain reasoning, check constraints, and jointly revise errors through interaction (Vygotsky and Cole, 1978; Rosé et al., 2008; Care et al., 2012). In contrast to product-oriented problem-solving processes, an exclusive focus on outcomes fails to capture how students actually collaborate through dialogue, including how

they negotiate meaning, consider different perspectives, generate solutions, and respond to one another in interaction. Although the importance of CPS as a process-oriented approach is well recognized in the mathematics literature (Roschelle and Teasley, 1995; O'Donnell and Hmelo-Silver, 2013; OECD, 2017), access to authentic classroom dialogue that enables fine-grained examination of these processes remains limited. Important educational dialogue resources such as Talk-Moves (Suresh et al., 2022), NCTE (Demszky and Hill, 2023), CIMA (Stasaski et al., 2020), and MathDial (Macina et al., 2023) are respectively more oriented toward classroom instructional discourse, tutoring dialogue, and constructed educational interaction, yet they do not provide dialogues centered on peer interactions in CPS.

To address this gap, we introduce PEERMATHDIAL, a CPS dialogue dataset collected from a math-centered, middle-school summer camp. The dataset captures students' naturally occurring face-to-face small-group problem solving and contains 55 unscripted dialogues from 27 students, comprising 6,406 dialogue turns in total. Our dataset preserves the spoken interactional process of classroom collaboration, including task interpretation, strategy proposal, peer questioning, etc. To the best of our knowledge, we are the first to publish a dataset of face-to-face peer CPS dialogues collected from authentic K-12 mathematics classrooms. Besides, to make the corpus more analyzable, we further derive a dialogue act (Stolcke et al., 2000) taxonomy to organize students' interactions. Rather than mapping dialogue turns onto expert-designed taxonomies such as the PISA CPS framework (OECD, 2017), we adopt an LLM-assisted, corpus-grounded inductive procedure: an LLM first processes the raw dialogue and summarizes dialogue actions, after which these candidate actions are manually reviewed by education experts. The resulting taxonomy covers major inter-

actional dimensions such as affective/off-task talk, communicating and coordinating, exploring and understanding, monitoring and revising, planning and executing, and representing and formulating. Its purpose is to provide a corpus-grounded organizational lens that makes subsequent educational dialogue analysis more tractable.

Beyond introducing the corpus and its action taxonomy, we demonstrate the applications of PEER-MATHDIAL through three specific applications. (1) **Understanding collaborative dynamics:** we track the temporal evolution of dialogues and empirically quantify the impact of teacher interventions on students’ subsequent micro-behaviors. (2) **Understanding the connection of actions and characters:** by aligning dialogue act distributions with self-reported character surveys (including math confidence, collaboration, leadership), we reveal typical patterns between students’ character and their behaviors in CPS. (3) **Evaluating LLM student simulation:** through dialogue act prediction experiments, we show that state-of-the-art LLMs still struggle to reproduce authentic collaborative behaviors, establishing our dataset as an ideal testbed for developing high-fidelity virtual student simulators.

2 Dataset

2.1 Data Collection

Our dataset was collected from a summer camp organized by George Mason University in 2025, centered on middle-school (i.e., grade 6-8) mathematical CPS. During each session, students participated in a 90-minute collaborative problem-solving block, in which they worked in small teams of 3-4 students on two to three mathematical tasks. To guide the session, the camp recruited 4 middle-school teachers from nearby school districts with an average of 20 years of teaching experience. The mathematical tasks were designed by these teachers, balancing between difficulty and engagement. Participants enrolled in the camp through recruitment materials distributed via the university’s camp website, social media platforms, and educational mailing lists. More details about the participants’ information and the camp tasks are in the Appendix A.

The instructional design emphasized open-ended, real-world-inspired mathematical tasks intended to elicit collaborative reasoning, mathematical modeling, and collective sense-making. Across

the week, students engaged in tasks that required them to propose solution strategies, justify their reasoning, and revise their ideas in response to peer feedback. For example, one task asked students to determine how to price a cake fairly based on its size under a fixed total price, requiring proportional and spatial reasoning. Together, these tasks were designed to foreground discussion, explanation, conjecturing, and revision rather than answer production alone.

All instructional sessions were video- and audio-recorded to capture both whole-class instruction and small-group interaction. We used Swivl cameras for whole-class video capture and placed an individual voice recorder at each table to obtain higher-quality audio of small-group collaborative dialogue. These recordings form the basis of the benchmark.

2.2 Data Processing and Statistics

To prepare the corpus, we maintained a centralized data log that tracked all available recordings by date, classroom, task, and group. We performed speech recognition using Otter AI.¹ All recordings were first transcribed automatically and then manually reviewed by four student researchers while listening to the original audio. We followed a two-level transcription workflow: (1) whole-lesson transcription, which captured teacher-led and class-wide discourse, and (2) small-group transcription, organized by table or group identity to preserve fine-grained peer interaction. Because Otter AI provides time-aligned transcripts, researchers were able to revise transcripts while checking the original recordings. We made light but systematic edits to improve lexical accuracy, punctuation, and readability, while preserving the original turn-by-turn structure of the interaction. Speaker identification was conducted manually. Otter AI initially assigned generic labels such as *Speaker 1* and *Speaker 2*; researchers replaced these labels with the corresponding student or teacher identities using session record sheets, group assignments, seating information, and audio/video context. During transcript cleanup, utterance segments that had been incorrectly split by the automatic system were merged to maintain speaker coherence and conversational flow. Real names were retained during the internal transcription stage to support cross-session tracking, and all transcripts were pseudonymized in

¹<https://otter.ai/>

Metric	Number
Total Student	27
Total Dialogues	55
Total Dialogue Turns	6,406
Average Turns per Dialogue	116.5
Average Words per Turn	11.6
Role Distribution	
Student	71.7%
Teacher	20.7%
Unknown	7.6%

Table 1: Statistics of the Dialogue Dataset.

later analytic stages. This processing pipeline was designed to preserve the temporal and interactional structure of collaborative dialogue, enabling analysis of participation, reasoning, and coordination in small-group mathematical problem solving.

The raw classroom transcripts contained overlapping whole-class instructional content. To only extract authentic peer interaction within small groups, we remove shared whole-class segments that were duplicated across multiple group transcripts, such as teacher-led openings at the beginning of class or summary discussions at the end. This process ensured that the final retained data consisted only of localized CPS dialogue within each group. In addition, we conducted a rigorous anonymization procedure, mapping original speaker names to consistent pseudonyms across all sessions. This allowed us to accurately track individual participation longitudinally while preserving privacy.

The statistics of the cleaned dataset are summarized in Table 1. The dataset contains 55 unique dialogue sessions with a total of 6,406 conversational turns. On average, each session includes 116.5 turns, and each turn contains 11.6 words, reflecting the relatively fast-paced and fragmented nature of middle school conversational exchanges. In terms of participation, the dataset tracks 27 focal students who completed self-report background surveys. Analysis of the role distribution indicates a student-driven CPS environment, with student utterances accounting for the vast majority of the discourse (71.7%). Finally, 7.6% of the turns are labeled as “unknown,” typically due to brief interjections, overlapping speech, or background noise that could not be reliably attributed to a specific speaker.

3 CPS Dialogue Act Discovery

Dialogue acts are functional units describing the speaker’s intent (Stolcke et al., 2000), which have

been widely used for discourse analysis and conversational system construction. Educational research on CPS discourse has often relied on pre-defined coding frameworks, such as the PISA framework (OECD, 2017; Stadler et al., 2020). These expert-developed frameworks typically focus on relatively coarse-grained behavioral categories, such as “establishing and maintaining shared understanding.” However, it cannot be directly applied to authentic middle school discussions. Student utterances are typically short and colloquial, and the granularity of the PISA framework is often too coarse for turn-level analysis. For example, questioning a problem condition and discussing a solution strategy may both be mapped to “establishing and maintaining shared understanding” under the PISA framework, despite serving different interactional dialogue act in local classroom discourse. To address this limitation, we adopt a bottom-up, LLM-assisted dialogue act induction pipeline that summarizes and organizes recurring interactional behaviors directly from the corpus. Table 2 summarizes the resulting taxonomy with details in Appendix B.

This pipeline is to leverage the LLM to identify the dialogue acts of each student turn within its full conversational environment. It consists of three steps. First, we divide the corpus into batches, each containing multiple dialogues, and submit the batch one by one to the LLM. For each batch, the model is asked to summarize the dialogue acts presented in this batch. Through this induction process, we extract repeated act descriptions across dialogues and gradually form a set of candidate dialogue acts. We then consolidate these batch-level summarized dialogue acts and explicitly control the granularity through prompting, so as to avoid acts that are overly broad (e.g., “establishing shared understanding”) or overly specific (e.g., “saying ok”), resulting in an initial act inventory tailored to this corpus. Second, after obtaining the initial act inventory, we ask the LLM to assign a dialogue act to every student turn in the dataset. For each turn, the model is required not only to provide an act label but also to produce a verbal confidence uncertainty score indicating how certain it is about the assignment (He et al., 2025). This allows us to identify the cases that the model itself considers uncertain. Third, we collect the turns with low confidence from the first round and submit them back to the LLM. At this stage, our goal is to examine whether these high-uncertainty cases reveal additional di-

Stage	Dialogue acts
Affect and Off-Task Talk	Off-Task or Affective Commentary
Communicating and Coordinating	Acknowledging or Receiving; Explaining or Justifying Reasoning; Guiding Peer Participation or Shared Workspace Actions; Presenting Group Reasoning to an External Listener
Exploring and Understanding	Asking for Clarification or Explanation; Identifying Relevant Elements and Constraints; Orienting to Lesson or Activity
Monitoring and Revising	Challenging or Diagnosing an Error; Checking What a Number or Step Represents; Checking Whether a Result Fits Constraints; Revising the Method or Answer
Planning and Executing	Carrying Out a Computation Step; Decomposing or Repartitioning the Structure; Proposing a Candidate Solution; Proposing a Solution Strategy
Representing and Formulating	Constructing or Refining a Representation; Mapping Representation to Values; Proposing a Pattern or Structural Model; Reusing a Previous Case or Pattern

Table 2: Overview of the corpus-grounded CPS dialogue act taxonomy. Full dialogue act definitions and examples are provided in Appendix B.

dialogue acts that are not yet covered by the initial act inventory. Through this focused inspection of low-confidence turns, the LLM can summarize potential new acts and thereby extend or revise the taxonomy. After obtaining the updated act inventory, we use the LLM to perform a new round of turn-level labeling over the full corpus. Overall, this pipeline combines corpus-level act discovery, low-confidence-case-driven taxonomy refinement, and full-corpus relabeling, thereby reducing the risk of missing dialogue act types as much as possible. All LLMs used are GPT-5.4 (OpenAI, 2026).

To evaluate the quality of the induced taxonomy and its turn-level labeling, we conducted an expert evaluation with two math education researchers who have multi-year experience as school teachers. The evaluation included two parts. In part 1, the experts were presented with the taxonomy definitions together with multiple examples and were asked if they agreed on them. In part 2, the experts evaluated 100 conversational turns, comprising 50 randomly sampled instances and 50 instances marked by the LLM as low-confidence. For each turn, we also provided the surrounding conversational context to disambiguate the judgment. The experts were then asked to judge whether the assigned act adequately described the student’s turn; if not, they were further asked whether a better label could be found in the taxonomy, or if the taxonomy was missing dialogue acts to cover the turn. As such, the experts were positioned to both *review the LLM assignment* and *make independent annotations*. Our results indicate that the dialogue act taxonomy provides strong coverage of the interactional behaviors in the corpus. The experts agreed

with the LLM’s assignments for all 50 randomly sampled turns and only identified 11 misclassifications of existing acts (rather than missing acts in the taxonomy) within the low-confidence set. Overall, the expert evaluation supports the validity of the induced taxonomy and suggests that this LLM-assisted pipeline is able to capture the major CPS interactional behaviors in the corpus with substantial coverage. We include details of the expert evaluation in Appendix C.

4 Application I: Enhancing Educational Understanding of the CPS Process

The discovered dialogue acts offer a lens through which educational researchers can better understand CPS. We present two possible directions in this section.

4.1 Temporal Evolution of CPS Dialogue Acts

We first investigate how the frequency of specific dialogue acts changes as a conversation unfolds. To compare dialogues of different lengths, we divide each conversation into six equal segments based on relative progress and compute dialogue-act proportions using student turns only. For each act, we first compute its proportion within each segment of each conversation and then average these proportions across conversations, preventing longer dialogues from dominating the estimate. For visualization, we express each segment’s value as a log-scaled deviation from that act’s average level across the full conversation. Under this normalization, positive values indicate greater-than-usual prominence, and negative values indicate lower-than-usual prominence.

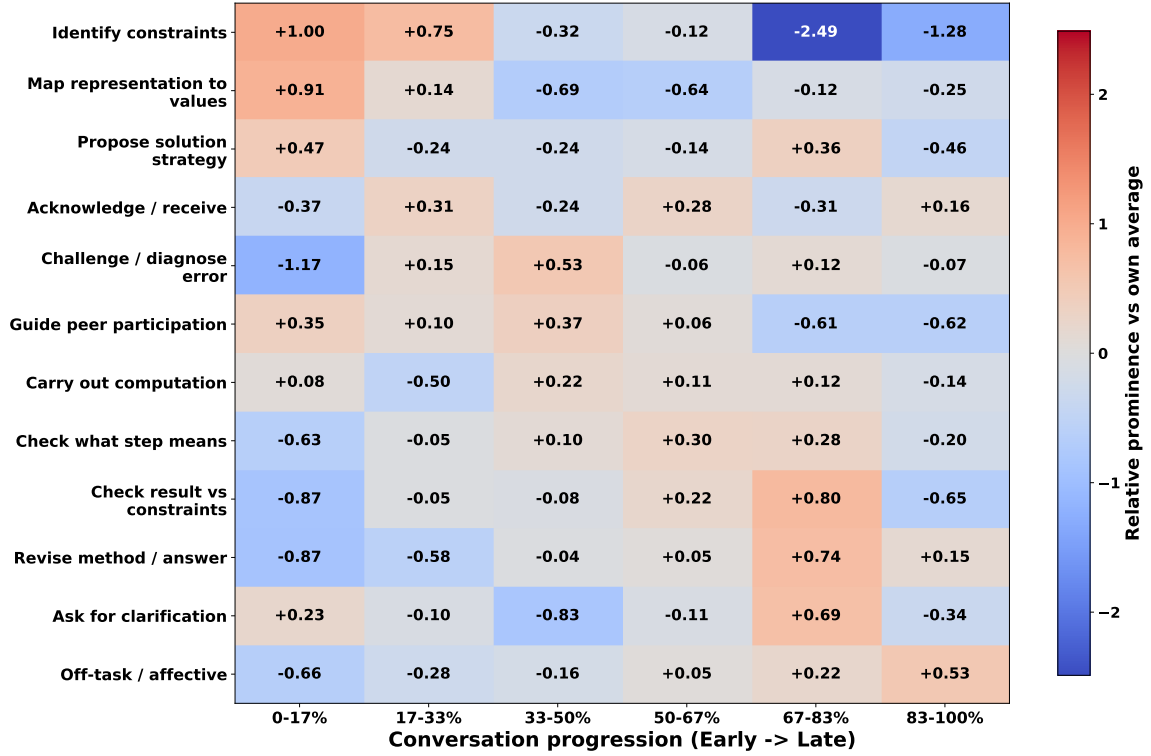


Figure 1: Temporal evolution of student dialogue acts over the course of CPS conversations. Each dialogue is uniformly divided into six stages based on relative progress. Let $p_{a,s}$ denote the proportion of act a within stage s , and $\bar{p}_a$ represent its overall average proportion across all stages. The heatmap visualizes the log-ratio, $\log_2(p_{a,s}/\bar{p}_a)$. Positive values indicate above-typical prominence, and negative values indicate below-typical prominence.

Figure 1 reveals a clear temporal structure in the CPS process when we align student turns by relative dialogue progress. Overall, the temporal pattern suggests that CPS dialogue is not static: it moves from early problem framing and coordination, to mid-stage calculation and error diagnosis, and finally to late-stage evaluation, revision, and convergence. Early stages are characterized by actions that help establish the problem space, such as *identifying relevant elements and constraints*, *mapping representation to values*, and *proposing a solution strategy*. At the same time, *guiding peer participation or shared workspace actions* is also more prominent near the beginning, suggesting that students initially devote effort to coordinating participation and setting up a shared working process. As the interaction moves toward the middle stages, the dialogue becomes more operational and diagnostic. Actions such as *challenging or diagnosing an error*, *carrying out a computation step*, and *checking what a number or step represents* become relatively more prominent, indicating a shift from task setup toward active solution development and local monitoring of intermediate reasoning. This middle portion appears to function

as the main problem-solving core, where students test ideas, compute, and interrogate the meaning of emerging results. In later stages, the discourse increasingly shifts toward evaluation and repair. *Checking whether a result fits constraints* and *revising the method or answer* become especially salient, consistent with a closing phase in which students verify whether their current solution is acceptable and make adjustments when necessary. Late-stage increases in *asking for clarification or explanation* suggest that final convergence often still requires resolving residual uncertainty rather than simply stating a completed answer. We also observe a modest rise in *off-task or affective commentary* toward the end, which may reflect easing task pressure once the main solution path has been established.

4.2 Impact of Teacher Intervention

Teacher intervention is an important component of CPS in educational settings, as teachers can redirect attention, clarify task demands, prompt reflection, and help groups move past local impasses. To better understand this role, we examine how students' dialogue acts shift immediately before and after

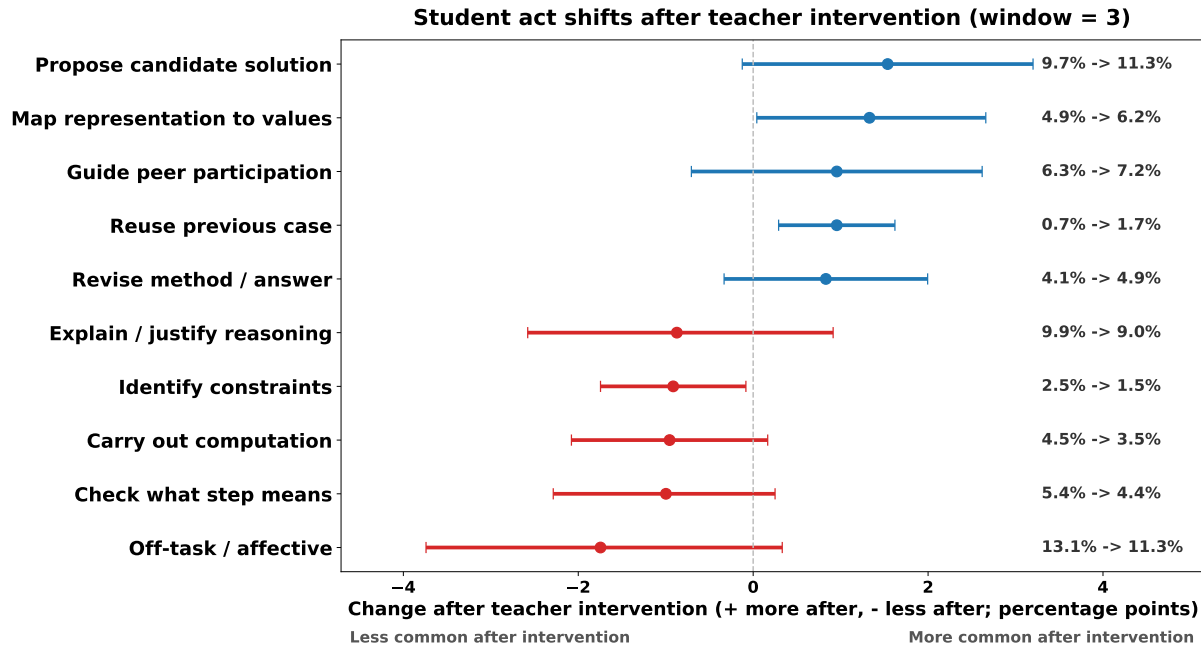


Figure 2: Change in student dialogue-act prevalence before versus after teacher intervention, using a three-turn student window on each side of the intervention.

teacher intervention. We operationalize each contiguous run of teacher turns as a single intervention event, and for each event, extract the three nearest annotated student turns immediately before it and three immediately after it. For every intervention event and every dialogue act, we compute the act's share in the pre-window and in the post-window, and define the event-level shift as *post share minus pre share*.

Figure 2 reports the mean of this post-minus-pre quantity across intervention, measured in percentage points: values to the right of zero indicate acts that become more common after teacher intervention, whereas values to the left indicate acts that become less common. Horizontal lines show 95% bootstrap confidence intervals. This setup allows us to characterize teacher intervention not as a single isolated response, but as a short-run redistribution of student activity in the turns that follow. The post-intervention shows reliable increases in *mapping representation to values* (from 4.9% to 6.2%) and *reusing a previous case or pattern* (from 0.7% to 1.7%), together with a reliable decrease in *identifying relevant elements and constraints* (from 2.5% to 1.5%). Descriptively, we also observe a shift toward *proposing a candidate solution* and away from *off-task or affective commentary*, although those trends are less certain because their confidence intervals overlap zero. The core takeaway is

that teacher facilitation appears to redirect the next few student turns away from local interpretation and toward forward-moving mathematical activity.

5 Application II: Facilitating Teacher Comprehension of Student Profiles

Beyond the understanding of the conversational process, dialogue acts serve as a powerful tool for understanding individual student characteristics. By aligning the dialogue acts distribution with self-reported surveys, we provide educators with a multi-dimensional view of student engagement. Student typologies are derived from a pre-program self-report survey. The survey consists of Likert-scale items with five response options ranging from *strongly disagree* to *strongly agree*, covering students' attitudes toward mathematics, collaboration, leadership, and related dispositions. For each profile dimension used in our analysis (e.g., confidence, collaboration, leadership), we group together the relevant survey items and assign students to one of two broad types based on whether their responses overall indicate endorsement of that dimension. In other words, students with more agreement-oriented responses on a given dimension are grouped into the corresponding type (e.g., *confident*, *collaborative*, or *leader-like*), whereas students without such endorsement are grouped into the contrasting type. Full survey items and

Most Interpretable Dialogue-Act Contrasts by Student Type

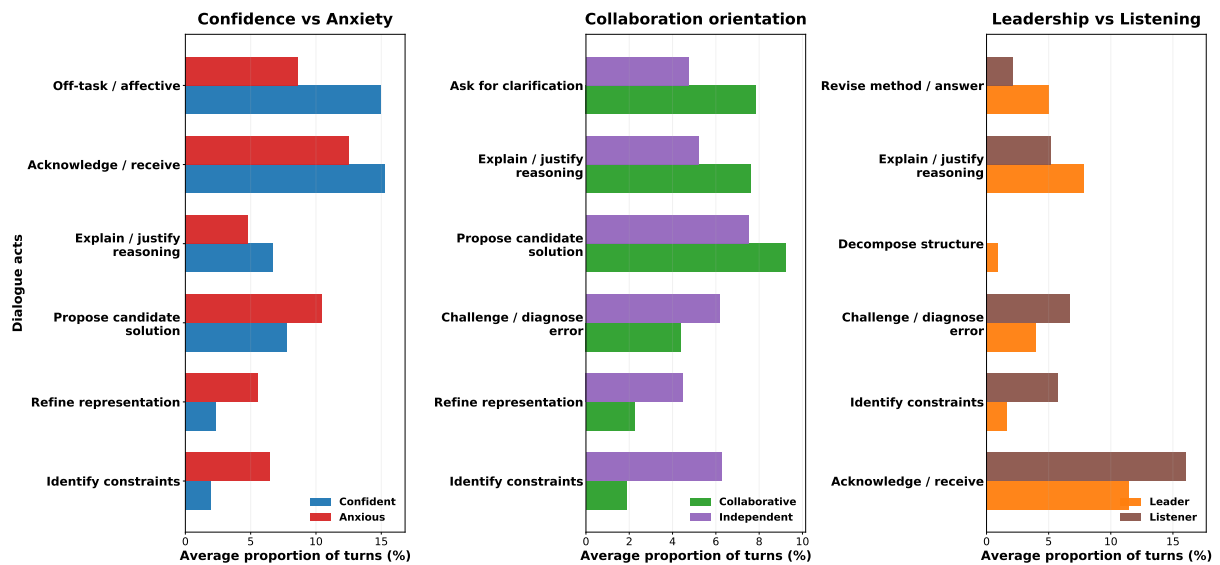


Figure 3: Comparison of dialogue-act usage. Students are grouped from pre-task survey responses along three dimensions: confidence (Confident vs. Anxious), collaboration orientation (Collaborative vs. Independent), and participation stance (Leader vs. Listener). Bar length shows the average proportion of each student’s annotated turns assigned to a given dialogue act.

their grouping are provided in Appendix D.

Dialogue Act Distributions Across Student Typologies Figure 3 shows the dialogue act distribution by student types. The results suggest that the survey dimensions capture differences in *how* students participate, rather than a single continuum of general participation. The clearest contrast appears along the leadership dimension. Compared with listeners, leader-type students produce more *revising the method or answer* (5.0% vs. 2.1%) and more *explaining or justifying reasoning* (7.9% vs. 5.2%). By contrast, listener-type students contribute more *acknowledging or receiving* (16.1% vs. 11.5%, difference favoring listeners by 4.58 percentage points) and more *identifying relevant elements and constraints* (5.7% vs. 1.6%). A similar functional differentiation appears along the collaboration dimension: collaborative students more often engage in *asking for clarification or explanation* (7.9% vs. 4.8%) and *explaining or justifying reasoning* (7.6% vs. 5.2%), whereas independent students more often foreground *identifying relevant elements and constraints* (6.3% vs. 1.9%). Along the confidence dimension, confident students show a higher rate of *off-task or affective commentary* (15.0% vs. 8.6%), while anxious students more often produce *identifying relevant elements and constraints* (6.5% vs. 1.9%, difference favoring

anxious students by 4.54 percentage points) and *constructing or refining a representation* (5.6% vs. 2.3%, difference favoring anxious students by 3.25 percentage points). Taken together, these patterns indicate that quieter or less leader-like students are not simply disengaged; rather, they tend to contribute through uptake, monitoring, and local grounding moves, whereas leader-like and collaborative students contribute disproportionately through explanation, clarification, and revision.

Discrepancies in Self-Reported Surveys At the same time, the dialogue-act analysis highlights an important limitation of treating self-reported survey responses as direct proxies for behavior. One particularly revealing case is the item “*I try to avoid arguments, even if I disagree.*” If interpreted literally, stronger agreement with this item might be expected to predict lower rates of *challenging or diagnosing an error*. However, the observed discourse does not follow that simple pattern. Among students with sufficient dialogue data, those who *agreed* or *strongly agreed* with the argument-avoidance item still produced *challenging or diagnosing an error* in 5.9% of their annotated turns, compared with 3.2% for students who gave a *neutral* response. One possible interpretation is that students understand “avoiding arguments” as avoiding interpersonal conflict rather

Table 3: Agreement between LLM-predicted actions and real student actions under naive student-type prompting, evaluated on 100 sampled cases.

Model	Match Rate
OpenAI GPT-5.4 Mini	16.0%
OpenAI GPT-5.4	18.0%
Anthropic Claude Sonnet 4.6	16.0%
Google Gemini 3 Flash Preview	20.0%
Qwen3.5-35B-A3B	17.0%

than refraining from mathematical critique. Another is that such behaviors are shaped by context: in relatively low-stakes tasks, students may perceive less social risk in pointing out an error. More broadly, this discrepancy underscores that survey responses capture self-perception and stated disposition, whereas dialogue acts capture situated behavior in collaborative activity. The two are related, but not isomorphic. For this reason, combining survey-based typologies with discourse-analytic evidence provides a more reliable account of student participation than either source alone.

6 Application III: Implications for LLM Student Simulation

In the field of educational AI, using LLM to simulate real students for interaction holds significant value (Yue et al., 2024). However, constructing high-fidelity student simulators remains highly challenging (Kumar et al., 2025). Virtual students must not only produce natural responses in dialogue, but their behavioral decisions must also align with the real students. PEERMATHDIAL provides a highly valuable ground truth for developing and evaluating such simulators.

To conduct an investigation of current LLMs’ ability, we designed an action prediction experiment. We randomly sampled 100 dialogue turns and provided the LLM with only the student survey and all utterances of the dialogue context. Without using complex scaffolding techniques, we asked the model to predict the most likely next action from the predefined dialogue act taxonomy. We evaluated five representative state-of-the-art models, including GPT-5.4, GPT-5.4-mini (OpenAI, 2026), Claude Sonnet 4.6 (Anthropic, 2026), Gemini 3 Flash Preview (Google, 2026), and the open-source Qwen3.5-35B-A3B (Qwen Team, 2026).

The experimental results in Table 3 indicate that current LLMs struggle to accurately reproduce the fine-grained collaborative behaviors of

real students when provided with only basic identity prompts and short context windows. Across all test cases, the exact match rate of predicted actions for each model ranged only between 16.0% and 20.0%. While we admit that in some contexts more than one dialogue act can be possible, this very low exact match rate still implies a mismatch between the LLM prediction and the majority of the dialogue development. This finding carries important implications for future educational applications of LLMs. If we aim to build virtual students with behaviorally realistic decision-making processes, relying solely on high-level persona attributes is insufficient. Future student simulators may require deeper fine-tuning, longer-term dialogue memory, or specialized modeling of mathematical collaboration states. The dataset we provide serves as an ideal testbed to support such advanced research directions.

7 Related Work

Collaborative Problem Solving (CPS) is a core educational competency in which learners build shared understanding, combine knowledge and skills, and work jointly toward a common goal (OECD, 2017). In mathematics education, open-ended tasks are widely recognized as supporting critical thinking and deep learning (Vygotsky and Cole, 1978), and prior work has explored how technological scaffolds can support social interaction and equitable participation (Rosé et al., 2008). Recent studies have also begun to use LLMs to support virtual collaborative environments, such as MathVC (Yue et al., 2024). However, these frameworks usually rely on the theoretical frameworks (OECD, 2017). In contrast, our work provides a corpus composed entirely of real human students and adopts a bottom-up inductive approach driven by LLMs to discover a dialogue act taxonomy that better reflects the authentic micro-processes of physical classrooms.

Datasets and LLM Applications for Education

Prior work has long explored conversational agents for supporting collaborative learning. For example, Kumar and Rose (2010) proposed an architecture for building conversational agents that support multi-learner collaborative learning settings. More recently, LLMs have been widely applied in education (Kasneci et al., 2023). The community has introduced several important dialogue datasets in recent years. These include TalkMoves (Suresh

et al., 2022), which focuses on analyzing classroom teacher instructional discourse; NCTE (Demszky and Hill, 2023), an elementary-school math-classroom dialogue dataset collected by the National Center for Teacher Effectiveness (NCTE); CIMA (Stasaski et al., 2020), designed to support tutoring dialogue generation; and MathDial (Macina et al., 2023), which simulates human teacher-student interactions grounded in math word problems. However, they focus on the interaction paradigm of “Teacher-Student”. Currently, there remains a severe lack of corpora and modeling efforts specifically targeting authentic horizontal “Peer-to-Peer” collaborative processes, and PEER-MATHDIAL fills this gap. In addition, our work showcases the potential of leveraging LLMs as powerful qualitative “analytical engines”, assisting educational researchers in automating discourse analysis and insight discovery.

8 Conclusion

We present PEERMATHDIAL, a new dataset of peer CPS dialogues from authentic middle-school mathematics learning environments, together with a corpus-grounded dialogue act taxonomy for analyzing collaborative interaction. Our experiments show that the dataset supports fine-grained study of collaborative dynamics, links between student traits and observed behaviors, and evaluation of LLM-based student simulation. We hope PEER-MATHDIAL will serve as a useful resource for future NLP and education research on collaborative learning processes.

9 Limitations

First, PEERMATHDIAL is collected from a single math-centered summer camp with a middle-school population, so the dataset does not fully represent the broader range of K-12 classrooms. Limited by the size and duration of the camp, the dataset we collected covers a limited number of students and dialogues. However, considering the lack of resources in peer interaction for math learning, our dataset still provides unique value to the community. Meanwhile, we encourage future studies to further examine our findings based on a larger population of peer interaction data. Second, our analyses and the scope of PEERMATHDIAL focus on spoken dialogue and do not fully capture other important aspects of collaborative problem solving, such as gesture, gaze, shared writing, and other multimodal

forms of coordination. This was mainly due to the technical difficulty in capturing synchronized multimodal data (e.g., recording students’ shared math plotting synchronized with their conversation). Finally, while we seek to demonstrate the broad applications of PEERMATHDIAL, each of the three applications is worth deeper exploration. For example, for LLM student simulation (Application III), future work should look into how the current simulation diverges from the dataset labels, whether this divergence is caused by the underspecification of student character or an inherent LLM bias, etc.

Acknowledgments

This work is supported by the National Science Foundation under award no. NSF-2418580 and NSF-2418582. We thank the Institutional Review Board (IRB) at George Mason University and William & Mary for the review and support of this research.

References

- Anthropic. 2026. Models overview. <https://platform.claude.com/docs/en/about-claude/models/overview>. Official Claude API documentation. Accessed: 2026-03-30.
- Esther Care, Patrick Griffin, and Barry McGaw. 2012. *Assessment and teaching of 21st century skills*. 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.
- Google. 2026. Gemini 3 developer guide. <https://ai.google.dev/gemini-api/docs/gemini-3>. Official Gemini API documentation. Accessed: 2026-03-30.
- Jianfeng He, Linlin Yu, Changbin Li, Runing Yang, Fanglan Chen, Kangshuo Li, Min Zhang, Shuo Lei, Xuchao Zhang, Mohammad Beigi, and 1 others. 2025. Survey of uncertainty estimation in llms-sources, methods, applications, and challenges. *Information Fusion*, page 104057.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, and 1 others. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

- Rohit Kumar and Carolyn P Rose. 2010. Architecture for building conversational agents that support collaborative learning. *IEEE Transactions on Learning Technologies*, 4(1):21–34.
- Sai Adith Senthil Kumar, Hao Yan, Saipavan Perepa, Murong Yue, and Ziyu Yao. 2025. Can llms simulate personas with reversed performance? a benchmark for counterfactual instruction following. *arXiv preprint arXiv:2504.06460*.
- Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621.
- OECD. 2017. [Pisa 2015 assessment and analytical framework: Science, reading, mathematic, financial literacy and collaborative problem solving](#).
- OpenAI. 2026. Models. <https://developers.openai.com/api/docs/models>. Official OpenAI API model documentation. Accessed: 2026-03-30.
- Angela M O’Donnell and Cindy E Hmelo-Silver. 2013. Introduction: What is collaborative learning?: An overview. *The international handbook of collaborative learning*, pages 1–15.
- Qwen Team. 2026. Qwen3.5-35b-a3b. <https://huggingface.co/Qwen/Qwen3.5-35B-A3B>. Official model card. Accessed: 2026-03-30.
- Jeremy Roschelle and Stephanie D Teasley. 1995. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, pages 69–97. Springer.
- Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. 2008. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International journal of computer-supported collaborative learning*, 3(3):237–271.
- Matthias Stadler, Katharina Herborn, Maida Mustafić, and Samuel Greiff. 2020. The assessment of collaborative problem solving in pisa 2015: An investigation of the validity of the pisa 2015 cps tasks. *Computers & Education*, 157:103964.
- Katherine Stasaski, Kimberly Kao, and Marti A Hearst. 2020. Cima: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Dan Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–374.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H Martin, and Tamara Sumner. 2022. The talkmoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. In *Proceedings of the thirteenth language resources and evaluation conference*, pages 4654–4662.
- Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.
- Murong Yue, Wenhan Lyu, Jennifer Suh, Yixuan Zhang, and Ziyu Yao. 2024. Mathvc: An llm-simulated multi-character virtual classroom for mathematics education. *arXiv preprint arXiv:2404.06711*.

A Student Demographic Information

The summer camp was held during the summer of 2025, supervised by the university IRB. In total, 38 students participated in the program. The camp was organized into two classrooms: one classroom with 16 sixth-grade students (except 1 fifth-grade talented-class student) and one classroom with 12 seventh-grade students and 10 eighth-grade students. The grade level was decided based on where they would enter in the 2025–2026 school year. The gender distribution included 26 female students (68%) and 12 male students (32%). Students' self-reported ethnic backgrounds included Asian or Asian American ($n = 16$), Black or African American ($n = 7$), White or European ($n = 6$), Hispanic or Latino/a ($n = 4$), Middle Eastern or North African ($n = 2$), Asian Indian ($n = 1$), White/European/Hispanic (multi-identified; $n = 1$), and prefer not to say ($n = 1$). Each classroom was facilitated by two teachers, all of whom had approximately 20 years of teaching experience on average. Student participants were also asked to fill out a survey form about their background and personality.

After data cleaning, we have 27 students who have both the survey information and the dialogue history. Therefore, we pick these 27 students' discussion in PEERMATHDIAL.

The dialogue data were collected from seven collaborative mathematics tasks: *Baker's Cake*, *Ice Cream*, *Znorlian*, *1–100*, *1001 Pennies*, *Tax Collector*, and *Traffic Jam*. These seven tasks correspond to 55 transcript sessions in total, including 11 sessions for *1001 Pennies*, 9 for *Baker's Cake*, 9 for *Znorlian*, 8 for *Ice Cream*, 8 for *Tax Collector*, 5 for *1–100*, and 5 for *Traffic Jam*.

B Action Taxonomy

This appendix presents the full corpus-grounded CPS dialogue act taxonomy used for turn-level annotation. The taxonomy is organized into six broad stages. Here, the term *stage* refers to a functional grouping of interactional moves rather than a fixed temporal sequence in collaboration. Each dialogue act is defined below, together with representative examples from the corpus.

B.1 Stage 1. Affect and Off-Task Talk

Stage definition. Actions used for emotional reactions, playful or evaluative commentary, and clearly social or off-task remarks that recur in classroom

discourse but do not directly advance the mathematical collaboration.

Stage 1 Action 1. Off-Task or Affective Commentary Definition. Making a brief affective reaction, playful remark, joke, evaluative exclamation, or clearly off-task social comment that does not primarily advance the mathematical work or direct collaboration.

Examples.

- “Well, clearly, he has stolen other people. He’s making me lose brain cells. He’s subtracting our brain power.”
- “Because... like two days ago, I went to Denmark. I mean, Delaware. Who Delaware.”
- “Oh, you’ve got to be kidding me.”
- “I thought we were going to be like coding and stuff, not math. This is basically new summer school.”
- “It’s me, hi. I know what the problem it’s you.”

B.2 Stage 2. Communicating and Coordinating

Stage definition. Actions used to explain and justify reasoning for others, present group work publicly, manage participation or work on the shared artifact, and handle lightweight interactional turns that support collaboration without adding substantive mathematical content.

Stage 2 Action 1. Acknowledging or Receiving Definition. Briefly accepting, confirming, or showing receipt of a prior contribution without adding substantial new mathematical content, such as short backchannels or uptake moves like “yeah,” “okay,” “oh,” or “uh-huh.”

Examples.

- “Uh, huh.”
- “Yeah.”
- “Okay.”
- “Yup.”
- “Oh, yeah.”

Stage 2 Action 2. Explaining or Justifying Reasoning **Definition.** Giving the rationale that connects a value, operation, representation, or method to the problem structure so others can follow why it makes sense.

Examples.

- “1.5 plus 1.5 would equal... because it’s half of the square, that’s 1.5 and not 2. And that means the entire square for one-fourth would have to be equal to four, but that’s not going to be four.”
- “That is three. That’s three because if it was split in half, this split in half. This would be six, but split in half again would be three.”
- “So over here, there are a bunch of cents. So there are, in one group of three, six numbers, 41, 41, 41, 41. 41 times four, 164, and 37 times two, 74, add it together equals 238 cents in one group of three. Well, one group of three pairs of numbers.”
- “It’s still half like the other one. That’s still half, and that’s still like each one fourth. So, nothing changed, except like the way it was shown.”
- “So, let’s say we have 19 cents, so that’s 19 cents, and then so we have 919 cents, so that’s \$9.19. So if we have 9919 cents then we have \$99.19.”

Stage 2 Action 3. Guiding Peer Participation or Shared Workspace Actions **Definition.** Directing who should contribute next or what should be drawn, written, compared, erased, or discussed on the shared workspace.

Examples.

- “Who’s next? Um, you.”
- “Okay, erase this and write plus that.”
- “You? So, you’re the scarecrow this time.”
- “All right, that’s one. Write that down.”
- “Can you write 228, 228.”

Stage 2 Action 4. Presenting Group Reasoning to an External Listener **Definition.** Summarizing the group’s process and answer for a teacher, class, verifier, or other audience outside the immediate back-and-forth collaboration.

Examples.

- “Okay, so, first we wrote a couple pennies here, and then we realized, started using the information we had to, like, organize, like which one is which. For example, we started with penny, nickel, dime, quarter, and then we realized, the second one, every two would be nickel. So, we first did that, and we filled out all the nickels here, and then we did every three would be dime. Then we realized they overlapped with the nickels, then Mrs. P told us that you’re going to have to replace them.”
- “We followed like the keys. So we just listed it out, like a bunch of pennies, and then we swapped out all the coins for like the numbers.”
- “I can go. So, and then we did, we did 1001 divided by 12, and we got 83, with five coins left.”
- “No, we realized, Mrs. P told us, what five coins. We added the first five coins, penny, nickel, dime, quarter, plus a penny, should equal 42. So, then we would add 42 here. At first, I thought this was the answer, then I realized the remainder. So we added 42 as the remainder and got 99.19.”
- “And then we added all the coins together, and we got 119 cents.”

B.3 Stage 3. Exploring and Understanding

Stage definition. Actions used to establish shared understanding of the task, lesson focus, givens, constraints, and the meaning of peers’ statements before or while the group commits to a mathematical path.

Stage 3 Action 1. Asking for Clarification or Explanation **Definition.** Requesting clarification about the task setup, a representation, an operation, or a peer’s reasoning so the group can align on understanding.

Examples.

- “Wait, how did we get two? How did we get two?”
- “How do you know that?”
- “Why are you putting a line?”
- “And then we’re putting it into, into. Are we dividing 1001 by that, or something?”
- “Where’d you get the 14 plus two?”

Stage 3 Action 2. Identifying Relevant Elements and Constraints **Definition.** Pointing out or restating key symbols, pieces, quantities, goals, rules, or structural givens so the group shares what must be tracked.

Examples.

- “First, let’s write down all the little, um, all the symbols we see. So I think we should an up, an up arrow. Triangle. And then this weird, like cross thing, and then, like a colored in square.”
- “Oh, yeah. So we have one, two, eight, and then we’re missing four.”
- “Yeah, cause the question says, what is the shortest way to write 21.”
- “It’s every third one.”
- “We’re not finding out these two groups specifically, we’re finding out how much cents are in those 1001 coins.”

Stage 3 Action 3. Orienting to Lesson or Activity **Definition.** Framing what kind of activity, topic, or phase is happening next, including teacher setup about what students are about to learn or do, rather than naming specific mathematical givens inside the problem.

Examples.

- “I feel like, for all these tasks, like, we have to, like, really understand the problem, and then we can solve it.”
- “The up is done. Adding numbers in Znorlian, and... is an addition sentence adding, what?”
- “So we have the exact same riddle. Easiest way to find 21.”
- “Okay. She said to do something. So I don’t know. Do you know that?”
- “Okay... now we have to find the shortest way to walk, write 21.”

B.4 Stage 4. Monitoring and Revising

Stage definition. Actions used to evaluate whether current reasoning fits the task, interpret intermediate quantities, locate errors, and adjust the approach or answer accordingly.

Stage 4 Action 1. Challenging or Diagnosing an Error **Definition.** Pointing out that a peer’s idea, representation, operation, or assumption is wrong and, when possible, identifying the specific source of the problem.

Examples.

- “Oh yeah, no cause it can’t be. It can’t be because this one would have to be. This, when you do it, this one’s three, and half of three is not two, it’s 1.5.”
- “No, because 12, minus two doesn’t equal four, that equals 10.”
- “That is half of one fourth, not a third.”
- “No, no, no, there’s no line in the middle.”
- “Well, how is one times 125 equal 185?”

Stage 4 Action 2. Checking What a Number or Step Represents **Definition.** Asking for or providing the meaning of an intermediate number, unit, or step to ensure quantities are being interpreted consistently.

Examples.

- “Huh, what’s 125? This equals 125? Does this equal 125?”
- “Two dollars and 38 cents.”
- “Two times two times two.”
- “Seven, but what are the values?”
- “The remainder of five coins.”

Stage 4 Action 3. Checking Whether a Result Fits Constraints **Definition.** Testing whether a proposed pattern, value, or answer is consistent with totals, fractions, repeating structure, or other task constraints.

Examples.

- “Okay so, this is um, um, um, what the heck, 1.5. Okay, so that’s 3.75. Okay, so now we add the 2.25, which should equal six. Right? That is correct. So your thing is correct.”
- “No, cause it wouldn’t add up to \$12.”
- “That’s three. That’s three. And then invisible line. Okay, that’s not even even. Nuh uh.”
- “So, we can’t have a half a coin?”
- “Because 5.5 would be ten... No, eleven. Plus three would be 14.”

Stage 4 Action 4. Revising the Method or Answer Definition. Changing the current answer, method, or shared written work after a contradiction, objection, or diagnosed error has been recognized.

Examples.

- “I’m gunna change my answer, three. It’s three. It’s been three the whole time.”
- “No, you’re not supposed to write this. You’re writing 16 plus four plus one.”
- “Uh, one third. No, one fourth.”
- “I’m changing the numbers because the math’s wrong.”
- “Yeah, there’s three different. Oh wait, yeah. There’s three different things, so I have to count for three? How many, we have to figure out groups of six, not groups of four.”

B.5 Stage 5. Planning and Executing

Stage definition. Actions used to choose solution methods, advance candidate answers, strategically reorganize the problem, and carry out computations.

Stage 5 Action 1. Carrying Out a Computation Step Definition. Performing or directing arithmetic or symbolic operations needed to execute the chosen strategy, including summing pattern units, dividing to find repeats, or combining known values.

Examples.

- “12 divided by four is three.”
- “So, two plus eight is ten, plus four is 14. So, something minus two is 14.”
- “1.5 plus 0.75, 2.25. That’s 2.25. That one’s 0.75.”
- “Eight times four is 32.”
- “No, this is two plus eight, which is ten, plus four is 14.”

Stage 5 Action 2. Decomposing or Repartitioning the Structure Definition. Redrawing, splitting, or grouping the representation into easier subparts to make inference or calculation more manageable.

Examples.

- “And then, here, you just put a line through. Just put a line through. That’s half of that. So, that will be 1.5 right?”
- “Okay, so then this, this would be three and three, but that can’t. So draw a line right here and right here. No right here, right here, just from here. No, no, no, here to here. Okay, so these are meant to be all equal squares. So these are all equal, and hold on can I be scarecrow for a sec? This... is three. This is all... three. So that is a... fourth, a fourth. What’s a fourth of three? I think that’s right, I don’t know. What’s a fourth of three? Go. A fourth of three.”
- “Okay, how do we split this exactly? Yeah, you should put it right here. There and... there’s another line, and just go right there. There. So, we know this is all six, and this all six. So, the easy part is, so this, it has to be... 0.75. So, 0.75 in here, and put imaginary lines over here. Wait. Do the uh... yeah. No, here first and then here. So this is 0.75, that’s 0.75, that’s 0.75. What are you doing exactly? Never mind. Okay, so that means this. This one, three. If we do that here too, it means 1.5. So, this is all 0.75, which would be three. This is three right here. So, put a three over here. So, this is a total of three. Oh, sorry. And then, this and three. This is all three minus 0.75, which would be um... 2.75. No, 2.25. So, this is 2.25.”
- “The bottom one is six. Okay, I know something. Okay, if we have. Little lines right here. It would be split in half. So it would be 1.5 because 1.5 plus 1.5 is three, and this would be like a quarter so it would be three. I’m guessing this is. So this would be. Is six. Basically.”
- “Okay so, okay so, put that in half. So, that we know is six. That’s six. Six!”

Stage 5 Action 3. Proposing a Candidate Solution Definition. Putting forward a tentative answer, value, decomposition, or completed expression for the group to consider.

Examples.

- “I think it should be \$4.”
- “16 plus five.”

- “Eight times three minus three.”
- “\$3 per slice.”
- “Oh, wait. I think I know it. Three. Three and 4.5. 4.5.”

Stage 5 Action 4. Proposing a Solution Strategy

Definition. Suggesting a method or plan for how to solve the problem, including how to decompose it, scale from a pattern, or proceed through the next mathematical steps.

Examples.

- “Yeah, we might be able to find the pattern, and then multiply.”
- “Okay, first things first, we’ll write what we know. Like, for example...”
- “Yeah, that’s where the pattern ends. The pattern ends at the 12 mark. If we add this, then we have to find how many times 12 goes into 1001. So we have to add this until like there, and we have to find out how many times that goes into 1001.”
- “What if we just like, what if we just like. So, we have 1001, right? Subtract how much amount of pennies will take over that. Every second one will be a one.”
- “So you have to do 250, divided by four, since 41, 41 appears every four different sections of the dollar.”

B.6 Stage 6. Representing and Formulating

Stage definition. Actions used to construct, interpret, and refine shared representations, patterns, and structural models that make the problem mathematically workable.

Stage 6 Action 1. Constructing or Refining a Representation

Definition. Creating, extending, or modifying a shared drawing, list, grouping, sample case, or auxiliary visual structure to make hidden relationships visible.

Examples.

- “And then, here, you just put a line through. Just put a line through. That’s half of that. So, that will be 1.5 right?”
- “Wait, I already, I already know it. So, square. Wait so, square right here, square right here.

There’s one right here, tall box right here, rectangular box right here, and there’s... oh wait. And then there’s. So then, there’s a line. Wait no, that’s not even. The line should be like right here. Three over here and three over here. So then, this is all, if this is all three then that means this is 1.5. How would it be two? So then, one of them would be 0.75. So, 0.75, 0.75. That means this also has to be 0.75. Plus this, which is right here. Wait, this line is supposed to be up here though. Let’s just draw this again. Half like this? Oh. And then, there’s a line right here. And then, there’s two boxes right here and a box right here. There! Wait, no. Yeah. So, this is 0.75, 0.75, don’t judge my fives. What did I just say. What did I just say. Trying my best here.”

- “Or maybe, to help you, we just put one, two, three, four, five, six.”
- “Okay. One, two, three, four, five, six, seven, eight, nine, ten. Okay, um, P, N, P, N, P, N, N, N. Okay, wait, okay, yeah, yeah. Okay, um...”
- “See, like that. Get it? Okay, and then quarters, every fourth one.”

Stage 6 Action 2. Mapping Representation to Values

Definition. Assigning numerical or relational meaning to symbols, shapes, pieces, or units based on givens and discovered relationships.

Examples.

- “One cent. Star is a nickel, five cents. Triangle is a dime, 10 cents. Heart is a quarter, 25.”
- “This symbol is a four. Um, this symbol is a one... uh, this symbol is an eight.”
- “One eighth. This is one eighth of the cake.”
- “A half.”
- “That means these mean one.”

Stage 6 Action 3. Proposing a Pattern or Structural Model

Definition. Suggesting a candidate pattern, grouping, equivalence, repeat unit, or structural interpretation that could organize the problem and guide solution work.

Examples.

- “So, the pattern is it doubles.”

- “This grouping four... and one, two, three, four, these four. Okay, then, then I see two different groups. One with these and one with these.”
- “That one’s four, that’s a 25. I think that’s where the pattern repeats.”
- “Oh. Two of these equals this, which equals that.”
- “So there are, basically, two different ones. There’s 41 and 37, and 41 and 41.”

Stage 6 Action 4. Reusing a Previous Case or Pattern Definition. Using an earlier solved example, repeated structure, or analogous case to interpret the current representation or infer the current one.

Examples.

- “I think it is. It’s the exact same thing, cause it’s in half. Three, three, six. It should be the same price cause there’s three slices, and then it’s doubled again.”
- “Oh. So, this is the exact same thing we get like with different shapes, but it’s still in half and... in half.”
- “Start drawing the square, right now. Oh wait, this is. Hold on, we recognize that right? That is half and half. Okay, so that is... half of six, three. So, three.”
- “It’s the same thing. That’s also 1.5. That’s 1.5. That’s six. That’s 4.5. It’s the same thing. It’s the same thing.”
- “Same thing. Same as day three.”

C Expert Review of Action Taxonomy and Turn-level Assignment

To evaluate the quality of the induced taxonomy and its turn-level labeling, we conducted an expert review. This evaluation focused on two questions. First, whether the dialogue act taxonomy induced by the LLM adequately captures the major dialogue actions that occur in students’ ordinary discussions; for this purpose, we provided the complete taxonomy definitions together with multiple examples. Second, whether the turn-level labels assigned by the LLM based on the taxonomy were accurate, and, if not, whether the problems arose because the taxonomy had failed to include a necessary action

type or because the appropriate action type existed but had been assigned incorrectly. Based on this goal, we constructed an evaluation set containing 100 instances of conversational turns. Among them, 50 instances were randomly sampled from the full corpus to assess labeling quality under typical conditions, and the other 50 instances were drawn from the turns that the LLM marked as low-confidence in the second round of labeling. This design allows the evaluation to cover both ordinary cases and the subset of examples that remain most uncertain under the final taxonomy and are therefore most likely to reveal residual labeling difficulties.

Two doctoral students who majored in Math Education and had multiple years of school teaching served as experts. During the review, each instance was presented to experts in a standardized format, including the speaker identity, the necessary dialogue context, the current turn to be judged, the action assigned by the LLM, and the model’s brief rationale. The experts were asked to judge whether the assigned act adequately described the student’s turn. If not, they were further asked whether a better label could be found among the existing candidate actions; if none of the existing acts fit, they were allowed to propose a new action label. In this way, the evaluation jointly examined both assignment quality and taxonomy coverage.

The results indicate that the taxonomy provides strong coverage of the major interactional behaviors in the corpus. The experts also endorsed the definitions of the extracted dialogue act taxonomy. Among the 50 randomly sampled instances, experts judged all assigned dialogue acts to be appropriate. Among the 50 low-confidence instances, 11 were judged to be assigned incorrectly. Importantly, in all of these error cases, the problem was identified as an assignment error rather than a missing act type in the taxonomy; that is, experts were always able to identify a relatively more suitable label from the existing candidate actions, without introducing any new category. This result suggests that the main remaining source of error lies in turn-level label assignment rather than insufficient coverage of the action inventory itself. Overall, the expert review supports the validity of the induced taxonomy and suggests that this LLM-assisted pipeline is able to capture the major CPS interactional behaviors in the corpus with substantial coverage.

D Survey-based Student Profiles

Student profile groupings used in Application II are derived from a self-report survey administered in the beginning of the camp program. The survey contains 20 items, each answered on a five-point Likert scale: *strongly disagree*, *disagree*, *neutral*, *agree*, and *strongly agree*. The items cover multiple aspects of students' orientations toward mathematics learning and peer collaboration, including math confidence and enjoyment, collaborative preference, leadership tendency, curiosity, organization, sociability, conflict avoidance, and concern about making mistakes. For the analyses in the main text, we organize related items into broader profile dimensions such as *confidence*, *collaboration*, and *leadership*. For each dimension, students are divided into two broad groups based on whether their responses indicate overall endorsement of that dimension (i.e., more agreement-oriented responses) or not. These groupings are intended as interpretable survey-based typologies for descriptive analysis, rather than as formal psychometric scales.

Survey items. The survey includes the following items:

- I enjoy learning math.
- I feel confident solving math problems.
- I often get frustrated when working on math.
- I like explaining math ideas to others.
- I find math to be useful in my daily life.
- I prefer working alone on math tasks.
- I feel comfortable asking questions in a group setting.
- Having others' perspectives often helps me learn better.
- Working in a team makes learning more engaging.
- I like being the center during the group discussion.
- I often listen to others before sharing my ideas.
- When I have an idea that others disagree with, I will persuade them.
- I often help assign tasks and resolve conflict in a team.
- I learn new things quickly.
- I'm a curious person and I like trying new things.
- I keep track of what needs to be done carefully.
- I like interacting with a lot of people.
- I try to avoid arguments, even if I disagree.
- I often worry about making mistakes in front of others.

Effects of Varying LLM Access on Essay Writing Behavior

Julia Christenson, Karin de Langis, Shirley Anugrah Hayati, Dongyeop Kang

University of Minnesota

{chri5306, dento019, hayat023, dongyeop}@umn.edu

Abstract

Investigating the degree to which large language models (LLMs) affect teaching and learning in universities can help identify strategies for integrating LLMs in a way that supports, rather than undermines, student learning outcomes. This study examined how varying levels of LLM assistance affect writing performance, engagement, and perceived authorship. We report a pilot study in which 24 college students were randomly assigned to write a short essay with no LLM access, limited access (≤ 3 prompts, responses capped at 100 words), or unlimited access. Overall essay quality was statistically indistinguishable across groups. Yet writing behavior and perceived authorship diverged sharply: students with limited access reported higher ownership (62.5% would submit the essay as independent work, vs. 25% in the unlimited group), stronger organizational gains, and more strategic, revision-focused prompting. The unlimited group spent more time writing, produced essays more similar to LLM output, and reported reduced creative expression. Our findings suggest that constraining, rather than banning, LLM access may preserve authorship confidence while retaining the scaffolding benefits of AI assistance.

1 Introduction

LLMs have become increasingly embedded in academic writing practices, reshaping how students plan, compose, and revise text. LLM-based tools such as ChatGPT and Grammarly now offer real-time feedback and idea generation, lowering barriers to writing while simultaneously raising questions about authorship, learning, and skill development. As these systems become normalized in educational contexts, it is important to understand their influence on student writing behavior.

Prior work primarily frames LLM-assisted writing as a productivity and accessibility aid, reporting gains in fluency, error reduction, and brainstorming support across literacy levels, as well as tools

that scaffold multiple stages of the writing process (Weijers et al., 2024; Lee et al., 2024; Zhang, 2025; Xu, 2025). At the same time, researchers caution that sustained reliance on LLM may undermine independent writing, creativity, and critical thinking (Rohilla, 2025; Gerlich, 2025a). Others find no significant effect on overall academic achievement (Yoo et al., 2025). Students often report reduced confidence and creativity with prolonged use, and prior work also highlights risks related to dependency, ethical ambiguity, and homogenized or biased outputs (Zhai et al., 2024; Chakrabarty et al., 2025). Human-AI interaction research further shows that AI involvement can negatively influence perceptions of authorship and writing quality, particularly when disclosed (Draxler et al., 2024; Li et al., 2024; Mansour et al., 2024; Joshi and Vogel, 2025; Gerlich, 2025b).

Existing research has shown that the use of LLMs has negative cognitive effects (Kosmyna et al., 2025), but we hypothesize that the degree of LLM access may play an important role in the efficacy of LLM assistants in educational contexts. This study addresses the gap by examining how different levels of LLM assistance influence students' writing by formulating three main research questions:

- **RQ1:** How does varied LLM assistance influence students' writing behavior, including time allocation, writing strategies, and prompting practices?
- **RQ2:** How do students perceive authorship, ownership, and acceptability of LLM-assisted writing?
- **RQ3:** How does the level of LLM assistance affect writing quality, originality, and lexical diversity in student essays?

To address these questions, we conduct a study in which twenty-four college students wrote an es-

say in response to a prompt. By comparing writing produced under different levels of LLM assistance, this work aims to clarify when LLM assistance enhances writing efficiency and when it begins to erode core literacy skills.

We found that both LLM-assisted groups engaged in similar write-prompt-edit cycles, but with different strategic orientations: limited-access participants focused prompts on major revisions and argument development, while unlimited-access participants concentrated on minor mechanical edits. These behavioral differences mapped onto perceived authorship: limited-access participants reported stronger feelings of ownership, greater organizational improvement, and higher willingness to submit the essay as their own work. Essay grades and lexical diversity were marginally higher in the unlimited group, though differences were not statistically significant. Taken together, the findings suggest that constraining LLM access, rather than eliminating it, may offer a middle path between the cognitive risks of over-reliance and the scaffolding benefits of AI assistance.

2 Related Work

Prior research has examined human-AI interactions for scholastic writing needs through engagement with LLMs and behavioral outcomes. Several studies on college students' use of LLMs find that learners primarily rely on LLMs for brainstorming, revision, and information retrieval (Jelson et al., 2025; Kondoro, 2025; Wang et al., 2024). Although LLM assistants effectively correct surface-level language errors, they have been shown to reduce syntactic complexity and linguistic diversity with sustained use (Wang and Spitz, 2025; Bui and Kim, 2025; Pan et al., 2025). Some studies found that students with higher AI literacy and more strategic prompting practices tend to produce stronger writing, while students with lower performance often under-utilize or misuse AI tools (Nguyen et al., 2024; Kim et al., 2025; Joshi and Vogel, 2025). A few studies observed modest improvements in students' independent learning and English writing ability when such tools are used alongside instruction (Wang, 2020; Seo, 2024; Xiao, 2024; Shen, 2025). However, these prior works share limitations. Most studies employ binary comparisons—AI versus no-AI—rather than examining a response relationship to varied levels of AI access. This framing obscures whether the negative effects

of LLM use are properties of AI assistance itself or the degree to which it is provided.

Growing concerns about dependence on LLMs negatively affecting critical thinking skills are prevalent. Survey research reveal many students and teachers are aware of the over-reliance on LLM writing assistants and concerns over academic dishonesty (Aliakbari et al., 2026; Joshi and Vogel, 2025; Kondoro, 2025; Khalid et al., 2025). Several studies show lowered critical thinking abilities for those who exhibit higher dependence on LLMs (Kosmyrna et al., 2025; Gerlich, 2025a).

Recent studies pivoted to researching methods to increase perceived ownership and transparency through tailored LLM assistants. One study has found that greater AI transparency and interpretability are associated with more positive student experiences and reduced misuse, including academic dishonesty and diminished learning outcomes (Cui, 2025). Several works have studied human-AI collaboration patterns and suggest writer-centered approaches to LLM-based writing assistants, (Mysore et al., 2025; Guo et al., 2024; Liu and August, 2025; Gerlich, 2025b; Shibani et al., 2024), but largely emphasize interaction strategies and design recommendations without measuring whether such interventions actually improve writing outcomes or reduce problematic dependency patterns.

Empirical studies demonstrate the benefits of LLM assistance, and the detrimental effects of AI on cognition and critical thinking skills, but the impact in real-world academic settings remains unclear (Yoo et al., 2025). Together, these gaps motivate our study focused on how constraining LLM access can impact how students navigate a real-world academic situation.

3 Approach

We aimed to investigate the extent to which students' writing experiences are impacted by the assistance LLM provides for them. The primary concern of the study was whether limiting LLM use provides a viable middle ground between the negative cognitive effects of LLM use and the benefits.¹

3.1 Participants

We recruited a total of 24 undergraduate and graduate students with 18 different majors through email recruitment. Details about participant demographic

¹The study was approved by the IRB.

Category	Percentage
Race	
White	41.7%
Asian	16.6%
Black or African-American	8.3%
Hispanic	4.2%
Two or more races	25.0%
Prefer not to disclose	4.2%
Gender	
Female	75.0%
Male	25.0%
Major	
STEM	41.7%
Health and Medicine	29.2%
Arts and Humanities	16.7%
Business	8.3%
Social Sciences	4.2%
Academic Level	
Senior	66.7%
Junior	16.7%
Sophomore	4.2%
Freshman	8.3%
Graduate	4.2%

Table 1: Participant demographics (N=24).

information are provided in Table 1. 71% of the participants rated themselves as strong writers by selecting either a 4 or 5 on a Likert scale out of 5. Participants, on average, have used LLM-based tools in some way in their previous 5 out of 10 writing assignments. Most students who use LLM tools in coursework reported using ChatGPT and built-in tools such as Grammarly. Participants who regularly use LLM-based tools typically employ them to generate ideas, seek assistance when they get stuck, and understand the material better. The participants who used LLM tools less frequently described the writing output as repetitive and superficial, and noted that they avoid LLM tools due to environmental concerns and academic integrity.

3.2 Study Procedure

First, participants completed a preliminary survey to determine eligibility. The eligibility requirements were native-level English fluency and identifying as an undergraduate or graduate student. Participants were randomly placed into one of three groups: control, limited access to LLM, or unlimited access. Participants then completed an approximately 30-minute in-person writing session. All participants provided informed consent under an IRB-approved protocol. Participants were given instructions to respond to an essay prompt in at least

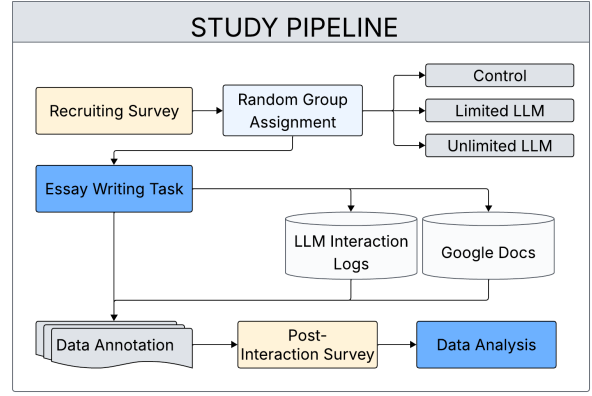


Figure 1: Study pipeline.

300 words, using a text document. Instructions can be found in Appendix A.1. Each participant received the same prompt, shown below:

Essay Prompt

Write a response in which you discuss the extent to which you agree or disagree with the recommendation and explain your reasoning for the position you take.

College students should be encouraged to pursue subjects that interest them rather than the courses that seem most likely to lead to jobs.

Since participants completed the writing task on their personal computers, we also instructed them to disable all AI writing extensions (e.g., Grammarly) on their browsers.

Participants in the control group did not receive any help from LLM when writing their essays, while participants in other groups have access to chat with LLM-based chatbots during their writing process. To access the LLM-based chatbot on the website, we implemented a backend application using Flask and Python, deployed with Vercel. We used the Gemini-2.5-Flash model (Gemini Team, 2025) for our chatbot because of our institution’s license.

Participants in the limited LLM access and unlimited LLM access groups interacted with their respective chatbots through a provided link. The writing assistant for the limited LLM group operated with a system prompt that restricted its responses to general guidance. The full system prompt is in Appendix A.2. This group was also given a maximum of three prompts to ask the LLM-based writing as-

Example Essay #1

Many people state that college is the best year of your life. I believe that is because it is a time for exploration and opportunities. This is the time to take new risks and try everything that interests you. However, students in college are at varying places in the world. Some have the privilege of getting financial help, whereas others may need to work multiple jobs to stay stable while getting their degree. While career readiness is important in a selection of classes, students should balance these with classes that also interest them.

Taking classes based on interest can fuel one's desire to learn and boost motivation. When students are actually curious about a topic, they will participate more in class and engage more deeply with the material. Connecting this to my own experience, I took a class on rock music, even though I am not studying music. I found myself enjoying the class a lot more because I enjoyed the topic and learning about music history.

Apart from student success in college, post-college happiness is often tied to fulfillment. Research shows that while income certainly matters up to a certain point (often cited around \$75,000 to alleviate daily stress), long-term job and life satisfaction are more closely tied to intrinsic factors like meaning and interest. These classes you can take in college are much more worth exploring if it could mean you find a new passion or even a new hobby.

LLM-assisted portion

Human-written portion

Figure 2: Example annotated essay with highlighted LLM-assisted portions for the limited group. This student interweaves responses from LLM to support their main claim in this essay.

sistant, whereas the unlimited access group had no such limit. The number of prompts was selected due to the shortened length of the writing task. The limited LLM group had a maximum token output and a system prompt instruction to provide short responses that reflect general guidance and encourage students to develop their own ideas, rather than longer responses that directly answer the prompt. Meanwhile, the unlimited LLM group did not have a restricted token output or special instructions.

Upon completion of the essay, participants were given a post-interaction survey to assess their perception of the writing process and their essay. Participants received a \$10 gift card as compensation.

4 Methodology

4.1 Data Collection & Annotation

We collected the interactions with the LLM via our web application and annotated the writing actions and interactions. Due to the limited number of participants and short writing task given, we developed a simplified taxonomy inspired by previous research (Du et al., 2022; Nguyen et al., 2024) with the following four writing intentions: writing, editing, structuring, and prompting.

We collected interaction data through a custom web application that logged participants' exchanges with the LLM, along with their writing activity in Google Docs. These logs enabled us to analyze both writing actions and prompting behavior throughout the essay-writing process.

To understand participants' writing behavior, we developed a taxonomy of four activities during the writing process: *writing*, *editing*, *structuring*, and *prompting*, informed by prior work on writing processes and revision behaviors. Writing refers to generating new text in full sentences, editing captures revisions to previously written content, and structuring involves organizing ideas or outlining the essay. Prompting means that the user is interacting with LLM to support writing. Two members of the research team manually annotated participants' writing intentions using this taxonomy, based on the Google Docs revision history and logged LLM interactions. Inter-annotator agreement for the writing intentions was assessed using Cohen's κ , yielding substantial agreement ($\kappa = 0.71$).

4.2 Metrics

We established a set of metrics designed to capture the nuances of writing behavior, essay quality, lexical/semantic overlap with LLM output, and vocabulary richness, allowing us to correlate them with survey responses later in the analysis. Given our limited statistical power to detect all but very large effects on writing quality, we present descriptive statistics without asserting statistical significance.

Rubric-Based Evaluation Essays were evaluated using a custom rubric derived from standard English composition criteria, as shown in A.2. The rubric assesses seven distinct dimensions: Focus and Details, Idea Support and Development, Organization, Voice, Vocabulary and Word Choice, Sentence Structure and Grammar, and Reads as human-written², in which a higher score meant more human-like quality. Each category was scored on a five-point scale and accompanied by a brief justification. This rubric was inspired by standard English essay rubrics (of Teachers of English).

To ensure reliability, grading was conducted independently by two university writing consultants employed at our institution's writing center, with the final rubric scores determined by averaging

²This metric functions as a holistic AI-detection proxy rather than a compositional quality measure

their two grades.³

Writing Time and Volume We tracked the total time each participant spent writing the essay (in minutes) to capture differences in writing effort and process efficiency across conditions. We also recorded the final word count of each essay as a measure of output length and productivity.

Lexical Diversity We measured lexical diversity using the Root Type-Token Ratio (Root TTR) (Guiraud, 1954). This metric provides a stable estimate of lexical diversity across essays of varying lengths by calculating the ratio of unique words (types) to the square root of the total words (tokens). Higher values indicate a broader, more diverse vocabulary usage relative to the total length of the text.

Prompt Type Prompts were categorized into four types: brainstorming/developing arguments, structure/organization, major revisions, and minor revisions. Developing arguments represented prompting to brainstorm talking points and specific facts or statistics. Structure/organization prompts focused on structuring the essay and making it cohesive. Major revisions included large revisions to the text, such as prompting to rewrite the entire essay, fixing grammatical mistakes, or modifying the essay stylistically. Lastly, minor revisions focused on rephrasing, sentence-level grammar checks, and asking for synonyms. Given the accuracy rate of annotations done by ChatGPT (Churchill et al., 2026), and due to the large number of prompts, the prompts were initially organized using GPT-4 (OpenAI et al., 2024) and all verified by one member of the research team with 94% accuracy.

AI Similarity Scores To assess lexical overlap with LLM output and the degree to which participants incorporated LLM-generated text, we computed cosine similarity scores between the final student essays and the raw LLM responses generated during their specific writing session. Sentence embeddings were generated using the all-MiniLM-L6-v2 model (Reimers and Gurevych, 2019). This resulted in a similarity score ranging from 0 to 1, where higher scores indicate a greater direct incorporation of LLM-generated phrasing.

³The graders were recruited via email and provided informed consent under an IRB-approved protocol.

5 Findings

As we attempt to understand how different levels of LLM assistance influence writing outcomes and writing behaviors, we organize our analysis around these main research questions: [RQ1]: How does varied LLM assistance influence students' writing behavior, including time allocation, writing strategies, and prompting practices? (Section 5.1) [RQ2]: How do students perceive authorship, ownership, and acceptability of LLM-assisted writing? (Section 5.2) [RQ3]: How does the level of LLM assistance affect writing quality, originality, and lexical diversity in student essays? (Section 5.3)

To address these questions, we combine rubric-based essay evaluation, lexical and similarity metrics, interaction logs, and survey responses. Each subsection below explicitly corresponds to one or more research questions.

5.1 Writing Behavior and Interaction Patterns

To examine how varied LLM assistance shapes writing behavior, we analyzed writing time, prompt usage patterns, and survey results across groups.

Writing Time Average writing time increased with greater LLM access. The control group completed essays in an average of 22.5 minutes, the limited LLM group in 27.8 minutes, and the unlimited LLM group in 30.4 minutes, as summarized in Table 5. Notably, variance also differed across groups: the unlimited group showed the least spread in completion time (most participants clustered near the 30-minute mark), while the limited group showed the greatest variability (Figure 3). One interpretation is that the three-prompt cap introduced individual differences in pacing: students who front-loaded prompts early finished faster, while those who saved prompts for revision extended their sessions. The unlimited group, facing no strategic constraint, may have converged on a similar write-and-revise rhythm regardless of individual differences. No correlation between time and essay grade was observed.

Writing Process We analyzed temporal writing intentions by normalizing each participant's writing session from start to finish. The limited group averaged 2.5 prompts, compared to 4.5 prompts for the unlimited group. Figures 5a, 5b, and 5c illustrate how participants distributed effort across planning, writing, prompting, and editing.

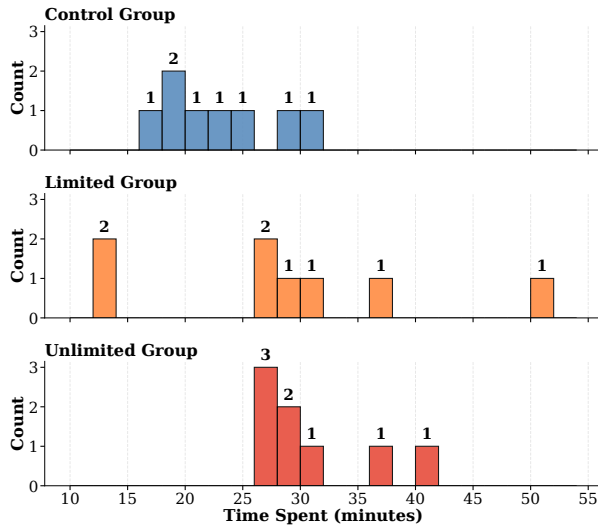


Figure 3: Distribution of writing time (in minutes) by user group.

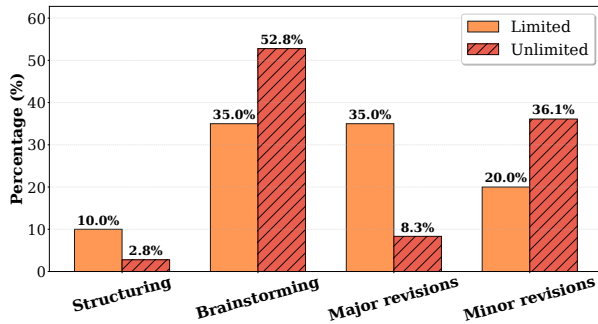


Figure 4: Distribution of the type of prompts asked by the LLM groups.

As shown in Figure 5a, two participants in the control group spent time on structuring the essay, and many participants spent most time writing sequentially, with additional time spent on editing the essay. Figure 5b shows the writing strategies of the limited LLM group. Six users demonstrated their collaboration with LLM, with writing themselves first, then prompting, and continuing this process until they were out of prompts. Despite having unlimited access to LLM, figure 5c shows that six of eight unlimited users still started the essay themselves, then prompted LLM for additional help. A majority of users in both groups with access to LLM utilized LLM in the latter part of their writing process (0.8-1).

Across all groups, most participants began by writing independently before seeking assistance. Editing behaviors were equally prevalent across all groups, while explicit structuring activities were relatively rare across all conditions. The increased LLM access was not associated with a difference

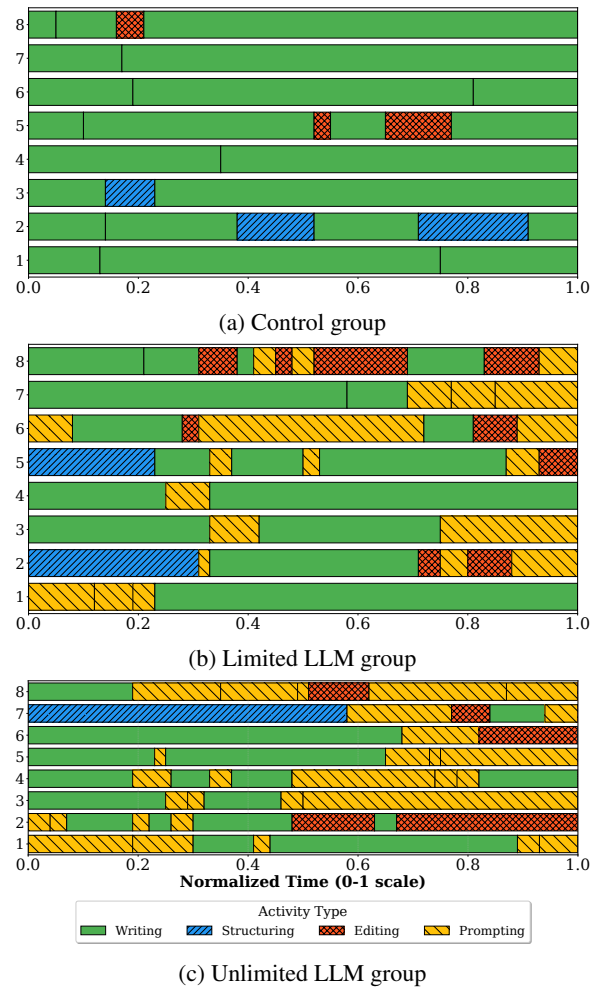


Figure 5: Distribution of writing intentions across the writing process for all user groups (8 participants per group). Each row represents a participant's timeline, with colored segments indicating their activities across normalized time.

in the writing and prompting process.

Prompt Types Figure 4 shows that brainstorming/argument development dominated prompts in both LLM assisted groups. Following argument-focused prompts, the unlimited group spent 36% of their prompts on minor revisions to the text. The limited users distributed prompts differently, with 35% on major revisions, and 20% on minor revisions.

Both groups showed low proportions of prompts directed toward structuring or organizing the essay. This pattern may reflect the possibility that participants may have completed structural planning independently, or the constrained length of the writing task may have reduced the concern over organization. The observed difference in major versus minor revision allocation between groups

Table 2: Limited group correlation matrix between pre-survey and post-session measures ($*p < 0.05$).

	Acceptance	Dev. Arguments	Time
Crit. thinking eval.	0.92*	—	—
Prompt literate	-0.80*	0.71*	—
Relied on AI	—	—	0.78*

Table 3: Unlimited group correlation matrix between pre-survey and post-session measures ($*p < 0.05$).

	Acceptance	Similarity	Prompt #
Crit. thinking eval.	—	-0.79*	—
Saved time	—	—	-0.71*
Comfortable with AI	-0.91*	—	—
AI use frequency	-0.80*	—	—
Difficulty	0.81*	—	—

aligns with expectations that the restricted number of prompts encouraged the students to prioritize higher-impact revisions.

5.2 Perceptions of Ownership and Acceptability

To understand students’ perceptions of authorship when using LLM to help their writing process, we analyzed responses from post-interaction surveys. We report the behavioral metrics as descriptive statistics due to low statistical power.

Post-interaction survey Post interaction responses revealed ambivalence toward LLM-assisted authorship.

First, participants in the limited group expressed more ownership towards their essays. When asked if they would submit it as their own work, 62.5% responded yes, 25% were unsure, and 12.5% declined. Most of the participants also perceived an increased ability to organize ideas (62.5%).

Correlational analysis within the limited group revealed several patterns (Table 2). Participants who used LLM frequently in writing assignments reported greater perceived need for critical thinking to decide what to accept or reject from the LLM ($r = 0.92, p < 0.05$). Those with self-reported prompt literacy accepted fewer prompts ($r = -0.80, p < 0.05$), but more frequently directed prompts towards developing arguments ($r = 0.71, p < 0.05$). This pattern suggests that the increased critical engagement with LLM outputs and prompting style may reflect an adjustment to the constrained nature of the system’s responses compared to their previous exposure to LLM tools, such as ChatGPT. Participants who reported spend-

ing more time writing the essay also revealed feeling greater reliance on LLM ($r = 0.78, p < 0.05$), suggesting more time was spent on the LLM interface, prompting, and incorporating suggestions.

Conversely, **the unlimited group demonstrated less ownership over their work**. Only 25% of these participants would submit their essay as their own work, and 75% claimed unsure. This suggests that this group had increased concerns over academic integrity and perceived originality. However, alternative explanations, such as low confidence in content or structure, cannot be ruled out.

Correlational analysis for the unlimited group (3) revealed several associations. A moderate negative correlation ($r = -0.79, p < 0.05$) between the cosine similarity score and the perceived critical thinking demands was observed. This implies that **students who actively evaluated and rejected LLM output produced more original work**, although causality cannot be determined.

Students in the unlimited group who believed they saved time using LLM used fewer prompts ($r = -0.71, p < 0.05$). Conversely, comfort with LLM tools ($r = -0.91, p < 0.05$) and higher baseline LLM use frequency ($r = -0.80, p < 0.05$) were associated with lower prompt acceptance rates. Additionally, perceived task difficulty showed a positive relationship with prompt acceptance ($r = 0.81, p < 0.05$), suggesting that students reporting higher difficulty more readily incorporated suggestions.

Notably, across both groups with access to LLM, students with stronger self-reported prompting skill accepted fewer suggestions overall (limited: $r = -0.80$; unlimited: $r = -0.91$). This pattern could indicate that **students with stronger prompting literacy applied more critical judgment in evaluating suggestions**. These results support prior research that prompt-literate individuals are more likely to think more critically of the output and engage in iterative interactions with LLMs (Kim et al., 2025).

Participants also expressed nuanced views on LLM use in coursework, aligning with previous studies (Aliakbari et al., 2026; Joshi and Vogel, 2025). Most responses indicated that LLM should be permitted for tasks such as brainstorming, source discovery, and concept clarification, but not for completing assignments or copying generated text. A small number of participants favored stricter limitations or unrestricted use, reflecting ongoing tension between perceived educational value and

Table 4: Average essay grades by user group, with highest scores in bold. Overall, differences between groups in grading outcomes are minor.

Rubric Item	Control	Limited	Unlimited
Focus + Details	4.3	4.5	4.6
Idea Support	4.6	4.5	4.7
Organization	4.3	4.3	4.7
Voice	4.9	4.7	4.5
Vocabulary	4.1	4.4	4.4
Sentence/Grammar	3.9	4.9	4.7
Reads as human-written	4.9	4.6	4.6
Total Score	88.66%	88.39%	89.38%

concerns about learning and academic integrity.

5.3 Writing Quality and Originality

To evaluate writing quality and originality, we analyzed essays using a rubric-based assessment, lexical diversity metrics, and similarity comparisons between student essays and LLM-generated responses.

Rubric Based Evaluation Table 4 shows the average rubric scores, which were largely stable between groups. The statistical power of our study only allows us to detect very large effects of group on grading, and we do not find any significant differences between groups.

Lexical Diversity As shown in Table 5, the unlimited group exhibited the highest average Root TTR (9.555), followed by the limited group (9.4365) and the control group (8.963). These results suggest that **access to LLMs may encourage broader vocabulary use**. However, we again cannot definitively determine any statistical relationships given our limited power.

AI Similarity Scores The unlimited group exhibited a higher average similarity score (0.552) compared to the limited group (0.444), suggesting marginally greater incorporation of LLM output (See Table 5). Although this is consistent with expectations, the difference was small and not statistically significant (ANOVA $p = 0.21$), preventing definitive conclusions regarding the dependence of LLM based only on similarity.

6 Discussion

Contrary to initial concerns, unlimited LLM access did not produce lower-quality essays than limited access or no LLM baselines. Only marginal differences in lexical diversity and similarity between

Table 5: Average Metric Evaluations by User Group

Metric	Control	Limited	Unlimited
Essay Grade	88.66%	88.39%	89.38%
Cosine Similarity	–	0.444	0.552
Root TTR	8.963	9.4365	9.555
Writing Time (mins)	22.5	27.8	30.4
Word Count	329	439	383

LLM suggestions and essay text were exhibited. However, the unlimited group spent more time writing and exhibited a higher similarity to LLM output (0.552 vs. 0.444), suggesting greater LLM reliance or slower, more iterative integration of LLM text. This aligns with prior work on cognitive offloading, where increased AI reliance may reduce rather than increase self-directed cognitive engagement (Gerlich, 2025a; Kosmyna et al., 2025).

The LLM groups demonstrated varied prompting styles. Both groups exhibited human-AI collaboration through iterating, prompting, and editing the essay, but the limited group prioritized major revisions, whereas the unlimited group targeted minor mechanical revisions. This strategic reallocation may have been correlated with measurable ownership benefits. 62.5% of limited participants reported willingness to submit their essay as independent work, compared to 25% in the unlimited group. Similarly, 62.5% of the limited group reported improved organizational ability.

Correlation analysis between pre- and post-interaction survey responses exhibited differences in how users interacted with the AI. As shown in Table 3, limited group participants who are comfortable with AI tools reported elevated critical thinking demands ($r = 0.92, p < 0.01$) and focused prompts on argument-development help ($r = 0.71, p < 0.05$). This pattern suggests that AI-literate students were more cautious and critical of the output. The unlimited LLM group exhibited a strong negative correlation between perceived need for critical thinking and LLM similarity scores. This suggests that students who evaluated LLM responses more rigorously produced more original work.

While constrained LLM access appeared to encourage strategic use and increase feelings of authorship, the study’s small sample size and simplified writing task limit generalizability.

7 Conclusion

This study examined how varied levels of LLM assistance influence college students' writing quality, writing behavior, and perceptions of authorship. In an IRB-approved study with 24 participants, results showed that overall essay quality showed no statistical significance, and systematic patterns in writing behavior, cognitive engagement, and ownership views revealed nuances about how LLMs impact student writing.

These findings motivate future research on LLM-assisted writing in more realistic academic settings, with larger samples and extended writing tasks, to better understand how students balance AI support with independent authorship.

8 Limitations

This study has several limitations. The data were collected over a two-week period from native English speaking college students in authors' university, which limits generalizability despite efforts to recruit participants from diverse backgrounds. The sample size was small, with 24 participants, due to time and resource constraints. As a result, the study focused on a single opinion-based essay prompt, which may not reflect writing behaviors or LLM interactions in other contexts, such as research-driven or evidence-based assignments. Longer writing tasks may reveal different patterns. Future work should explicitly examine EFL/ESL learners, who represent a significant and potentially more LLM-reliant group.

The model also lacked built-in conversational memory. Participants assumed that the LLM would retain context across prompts, leading to confusion when follow-up prompts were interpreted in isolation. This issue was only identified after receiving feedback from the initial participants.

Additional limitations arose in data collection and annotation. Google Docs editing histories only captured major revisions, preventing fine-grained analysis of sentence-level edits and micro revisions during continuous writing. Manual annotation of writing intentions, while validated through inter-rater agreement ($\kappa = 0.71$), introduced potential bias. Additionally, we used Gemini-2.5-Flash rather than ChatGPT, which may limit comparison to students' real-world AI experiences. We did not observe any clear attempts to circumvent the system prompt constraints. Alternative explanations for correlational outcomes (e.g., low confidence

rather than AI-induced ambiguity) cannot be fully ruled out.

We also used cosine similarity to determine an AI similarity score, which captures surface-level semantic overlap but does not directly measure originality in a deeper cognitive sense.

Future work should expand participant pools across institutions and regions, include multiple writing tasks of varying genres and lengths, incorporate memory-enabled frameworks to better reflect real-world AI writing tools, and employ more detailed logging mechanisms to capture granular writing behavior. Longitudinal designs examining skill development would also help capture long term effects of varied LLM access on writing ability.

9 Ethical Considerations

This study does have ethical considerations related to participant privacy, informed consent, and responsible use of generative AI. All participants were fully informed of the purpose of this study and the structure of the writing tasks. Participation was voluntary, and students were free to withdraw at any time without penalty. Incentives were given to participants as per gift cards in return for their time and effort. The study also considered the ethical implications of exposing participants to biased, inaccurate, or misleading AI-assisted output. The LLM was constrained to provide short, general writing suggestions rather than fully generated paragraphs or factual claims. Our study does have potential harm if there is a misinterpretation of our findings, which could encourage the misuse or altogether elimination of AI tools in a school environment. To address this, we emphasized the balance, acknowledgment, and responsibility of AI use, especially in an academic setting.

References

- Mohammad Aliakbari, Pooria Barzan, and Seyyed Pedram Allahveysi. 2026. [AI usage in academic writing: Perspectives of stakeholders](#). *AI and Tech in Behavioral and Social Sciences*, 3(4):1–12.
- Minh Ngoc Bui and Eun-young Kim. 2025. [Analysis of influential factors on the writing quality of chatgpt-generated academic essays](#). In *Proceedings of the 2024 7th International Conference on Educational Technology Management, ICETM '24*, page 315–321, New York, NY, USA. Association for Computing Machinery.

- Tuhin Chakrabarty, Philippe Laban, and Chien-Sheng Wu. 2025. [Can ai writing be salvaged? mitigating idiosyncrasies and improving human-ai alignment in the writing process through edits](#). In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.
- Alexander Churchill, Shamitha Pichika, Chengxin Xu, and Ying Liu. 2026. [Gpt models for text annotation: An empirical exploration in public policy research](#). *Policy Studies Journal*, 54(1):e70034. E70034 PSJ-24-RNR-0078.R2.
- Yulu Cui. 2025. [What influences college students using ai for academic writing? - a quantitative analysis based on hisam and tri theory](#). *Computers and Education: Artificial Intelligence*, 8.
- Fiona Draxler, Anna Werner, Florian Lehmann, Matthias Hoppe, Albrecht Schmidt, Daniel Buschek, and Robin Welsch. 2024. [The ai ghostwriter effect: When users do not perceive ownership of ai-generated text but self-declare as authors](#). *ACM Trans. Comput.-Hum. Interact.*, 31(2).
- 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, Dublin, Ireland. Association for Computational Linguistics.
- Google Gemini Team. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.
- Michael Gerlich. 2025a. [Ai tools in society: Impacts on cognitive offloading and the future of critical thinking](#). *Societies*, 15(1).
- Michael Gerlich. 2025b. [From offloading to engagement: An experimental study on structured prompting and critical reasoning with generative ai](#). *Data*, 10(11).
- Pierre Guiraud. 1954. Les caractères statistiques du vocabulaire: essai de méthodologie. *Presses universitaires de France*.
- Alicia Guo, Leijie Wang, Jeffrey Heer, and Amy Zhang. 2024. [Preserving writer values in ai writing assistance tools](#). In *Proceedings of the Third Workshop on Intelligent and Interactive Writing Assistants*, In2Writing '24, page 58–61, New York, NY, USA. Association for Computing Machinery.
- Andrew Jelson, Daniel Manesh, Alice Jang, Daniel Dunlap, Young-Ho Kim, and Sang Won Lee. 2025. [An empirical study to understand how students use chatgpt for writing essays](#). *Preprint*, arXiv:2501.10551.
- Nikhita Joshi and Daniel Vogel. 2025. [Writing with ai lowers psychological ownership, but longer prompts can help](#). In *Proceedings of the 7th ACM Conference on Conversational User Interfaces*, CUI '25, page 1–17. ACM.
- Eman Khalid, Hammad Javaid, Yashal Waseem, and Natasha Sohail Barlas. 2025. [AI as a mind partner: Cognitive impact in Pakistan's educational landscape](#). In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Works in Progress*, pages 14–19, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- Jinhee Kim, Seongryeong Yu, Sang-Soog Lee, and Rita Detrick. 2025. [Students' prompt patterns and its effects in ai-assisted academic writing: Focusing on students' level of ai literacy](#). *Journal of Research on Technology in Education*, pages 1–18.
- Alfred Malengo Kondoro. 2025. [AI writing assistants in tanzanian universities: Adoption trends, challenges, and opportunities](#). In *Proceedings of the Fourth Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2025)*, pages 37–46, Albuquerque, New Mexico, US. Association for Computational Linguistics.
- Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. 2025. [Your brain on chatgpt: Accumulation of cognitive debt when using an ai assistant for essay writing task](#). *Preprint*, arXiv:2506.08872.
- Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou, Emad A. Alghamdi, Tal August, Avinash Bhat, Madiha Zahrah Choksi, Senjuti Dutta, Jin L.C. Guo, Md Naimul Hoque, Yewon Kim, Simon Knight, Seyed Parsa Neshaei, and 17 others. 2024. [A design space for intelligent and interactive writing assistants](#). In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.
- Zhuoyan Li, Chen Liang, Jing Peng, and Ming Yin. 2024. [How does the disclosure of AI assistance affect the perceptions of writing?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4849–4868, Miami, Florida, USA. Association for Computational Linguistics.
- Yijun Liu and Tal August. 2025. [From crafting text to crafting thought: Grounding intelligent writing support to writing center pedagogy](#). In *Proceedings of the Fourth Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2025)*, pages 47–61, Albuquerque, New Mexico, US. Association for Computational Linguistics.

- Atil Mansour, Ofra Amir, and Liat Levontin. 2024. [The potentially negative effects of ai writing assistants on self-disclosure](#). In *Proceedings of the Third Workshop on Intelligent and Interactive Writing Assistants, In2Writing '24*, page 1–3, New York, NY, USA. Association for Computing Machinery.
- Sheshera Mysore, Debarati Das, Hancheng Cao, and Bahareh Sarrafzadeh. 2025. [Prototypical human-AI collaboration behaviors from LLM-assisted writing in the wild](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16830–16857, Suzhou, China. Association for Computational Linguistics.
- Andy Nguyen, Yvonne Hong, Belle Dang, and Xiaoshan Huang. 2024. [Human-ai collaboration patterns in ai-assisted academic writing](#). *Studies in Higher Education*, 49(5):847–864.
- National Council of Teachers of English. [Essay rubric](#).
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Hongchen Pan, Eduardo Araujo Oliveira, and Rafael Ferreira Mello. 2025. [Exploring human-ai collaboration in educational contexts: Insights from writing analytics and authorship attribution](#). In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK '25*, page 903–909, New York, NY, USA. Association for Computing Machinery.
- 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.
- Atiya Rohilla. 2025. [Impact of excessive ai tool usage on the cognitive abilities of undergraduate students: A mixed method study](#). *Applied Science and Sustainable Agriculture Journal*, 4(1):2131–2143.
- Ji-Young Seo. 2024. Exploring the educational potential of chatgpt: Ai-assisted narrative writing for efl college students. *Language Teaching Research Quarterly*, 43:1–21.
- Yixin Shen. 2025. [Ai-driven scaffolding and affective support in esl argumentative writing: A multimodal analytics approach](#). In *Proceedings of the 2025 6th International Conference on Education, Knowledge and Information Management, ICEKIM '25*, page 604–608, New York, NY, USA. Association for Computing Machinery.
- Antonette Shibani, Simon Knight, Kirsty Kitto, Ajanie Karunanayake, and Simon Buckingham Shum. 2024. [Untangling critical interaction with ai in students' written assessment](#). In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI EA '24*, New York, NY, USA. Association for Computing Machinery.
- Chaoran Wang, Zixi Li, and Curtis Bonk. 2024. [Understanding self-directed learning in ai-assisted writing: A mixed methods study of postsecondary learners](#). *Computers and Education: Artificial Intelligence*, 6:100247.
- Mengying Wang and Andreas Spitz. 2025. [Quantifying the risks of LLM- and tool-assisted rephrasing to linguistic diversity](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22561–22574, Suzhou, China. Association for Computational Linguistics.
- Zhijie Wang. 2020. [Computer-assisted efl writing and evaluations based on artificial intelligence: a case from a college reading and writing course](#). *Library Hi Tech*, 40(1):80–97.
- Ruben Weijers, Gabrielle Fidelis de Castilho, Jean-François Godbout, Reihaneh Rabbany, and Kellin Pelrine. 2024. [Quantifying learning-style adaptation in effectiveness of LLM teaching](#). In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 112–118, St. Julians, Malta. Association for Computational Linguistics.
- Qimin Xiao. 2024. [Chatgpt as an artificial intelligence \(ai\) writing assistant for efl learners: An exploratory study of its effects on english writing proficiency](#). In *Proceedings of the 2024 9th International Conference on Information and Education Innovations, ICIEI '24*, page 51–56, New York, NY, USA. Association for Computing Machinery.
- Ziyang Xu. 2025. [Patterns and purposes: A cross-journal analysis of ai tool usage in academic writing](#). *Preprint*, arXiv:2502.00632.
- Tae-An Yoo, Min-Kyu Kim, Ho-Suk Yang, and Ji-Bum Chung. 2025. [The impact of ai-assisted writing on students' learning effectiveness and perceptions: A mixed-methods study](#). *SSRN Electronic Journal*. Available at SSRN: <https://ssrn.com/abstract=5396600>.
- Cheng Zhai, Setio Wibowo, and Li Da Li. 2024. [The effects of over-reliance on ai dialogue systems on students' cognitive abilities: a systematic review](#). *Smart Learning Environments*, 11(28).
- Tingting Zhang. 2025. [Evaluating the impact of chatgpt-assisted writing: A controlled study on undergraduate academic compositions](#). In *Proceedings of the 2025 6th International Conference on Education, Knowledge and Information Management, ICEKIM '25*, page 258–262, New York, NY, USA. Association for Computing Machinery.

A System Design Details

A.1 Study Procedure

Participants were recruited via email and asked to complete a preliminary survey administered through Google Forms. Upon receipt of the completed survey, participants were scheduled for an individual, in-person study session.

Each session took place in a private room. Participants were first asked to read and sign the informed consent form (Appendix A.1.1) before beginning the study. Following consent, participants received an email containing two links: one to a document outlining study instructions and another to a Google Doc for composing their essay. Participants assigned to the limited-AI and unlimited-AI conditions additionally received login credentials via email to access the AI-assisted writing platform. The full set of condition-specific instructions is provided in Appendix A.1.2.

All participant data were stored in a private Google Drive folder accessible only to members of the research team. Identifying information was removed from all essays and replaced with anonymized user IDs. During data collection through the LLM interface, only anonymized user IDs were recorded to protect participant privacy.

A.1.1 Consent Form

You are invited to participate in a research project investigating the Overdose Effect of AI-Assisted Writing at the University of Minnesota - Twin Cities. We ask that you read this form and ask any questions you may have before agreeing to take part.

Purpose of the Study: This project explores how different levels of reliance on LLM tools (such as ChatGPT or other generative AI systems) affect students' writing ability, creativity, and critical thinking. The goal is to understand how LLM assistance may influence the quality, style, and originality of written work.

Procedures: The participant will engage in a short preliminary survey asking about their background, experience with AI tools, and comfort level using them. We will then have the participant write a short essay (at least 250-300 words) in response to a provided prompt within approximately 30 minutes. Lastly, the participant will be asked to complete a short post-survey reflecting on their writing process and any use of AI during the activity.

Risks and Benefits of Being in the Study: There

are minimal risks to being in the study. While there is no direct benefit to you, this study will help us better understand how AI tools impact writing skills and academic performance. Each user will be compensated with a \$10 gift card for their time.

Data Sharing: The information you provide through this project will be recorded. Data that you provide as part of this research project will be shared with the other students on the project team, and the course staff.

Confidentiality: We will not collect any information that will make it possible to trace your participation back to you. We will not share your participation with anybody outside of the project team. We will keep your participation private to the extent allowable by law.

Voluntary Nature of Research Participation: Participation in this project is voluntary. Your decision whether or not to participate will not affect your current or future relations with the University of Minnesota - Twin Cities.

Contacts and Questions: The students conducting this research project are: Julia Christenson, Shirley Hayati, and Karin de Langis. The instructor supervising this project is Dongyeop Kang. You may ask any questions you have now. If you have questions later, you are encouraged to contact us. You may keep this page for your records after signing and returning the attached sheet.

Statement of Consent: I have read the attached information regarding the project investigation of the Overdose Effect of AI-Assisted Writing at the University of Minnesota - Twin Cities. I have asked questions and have received answers. I consent to participate in this project.

A.1.2 Instruction Forms

Control Group "A Google Doc will have been shared with you for you to write your essay in. Write a response to the prompt below. Your response should be around 300 words (at least 3 paragraphs), however, you can write more if you want to. Take as much time as you need to write your response.

Prompt: Write a response in which you discuss the extent to which you agree or disagree with the recommendation and explain your reasoning for the position you take. *College students should be encouraged to pursue subjects that interest them rather than the courses that seem most likely to lead to jobs.*

Once you are done writing your response fill out

the following survey. Once you are finished with the essay and the survey please let one of the study team members know."

Limited Group "A Google Doc will have been shared with you for you to write your essay in. Write a response to the prompt below. Your response should be around 300 words (at least 3 paragraphs), however, you can write more if you want to. Take as much time as you need to write your response.

When writing your response, **you are allowed to prompt AI 3 times**. You don't have to use all or any of your allotted prompts; however, you cannot use more than 3. You also cannot prompt the AI to write the full response for you. To access AI, go to this link and sign in with the username and password that the study team member gave you.

Prompt: Write a response in which you discuss the extent to which you agree or disagree with the recommendation and explain your reasoning for the position you take. *College students should be encouraged to pursue subjects that interest them rather than the courses that seem most likely to lead to jobs.*

Once you are done writing your response fill out the following survey. Once you are finished with the essay and the survey please let one of the study team members know."

Unlimited Group "A Google Doc will have been shared with you for you to write your essay in. Write a response to the prompt below. Your response should be around 300 words (at least 3 paragraphs), however, you can write more if you want to. Take as much time as you need to write your response.

When writing your response, **you are allowed to prompt AI as many times as you would like** to help you write your response. However, you cannot prompt the AI to write the full response for you. (You cannot copy and paste in the prompt.) To access AI, go to this link and sign in with the username and password that the study team member gave you.

Prompt: Write a response in which you discuss the extent to which you agree or disagree with the recommendation and explain your reasoning for the position you take. *College students should be encouraged to pursue subjects that interest them rather than the courses that seem most likely to lead to jobs.*

Once you are done writing your response fill out the following survey. Once you are finished with

the essay and the survey please let one of the study team members know."

A.2 Study Interface

The full set of instructions for the limited AI model before the user's prompt is below.

```
SYSTEM_PROMPT = "You are a helpful study assistant. Your role is to provide general guidance, suggestions, and help students think through their work. IMPORTANT RULES: - Keep ALL responses under 100 words - be concise and focused - Provide suggestions, outlines, and general guidance only - DO NOT write complete essays, paragraphs, or full answers that students can copy-paste - Help students develop their own ideas through questions and prompts - Offer structural advice (e.g., "Consider organizing your essay with: intro, 3 body paragraphs, conclusion") - Suggest topics to research or think about - Ask clarifying questions to help students think deeper If asked to write a complete essay or full answer, politely decline and instead offer to: 1. Help them brainstorm ideas 2. Suggest an outline structure 3. Provide tips on how to approach the topic 4. Ask questions to help them develop their thoughts"
```

The interactions with the chatbot and user were sent to a csv file using Vercel Blob. Each row of data followed this order: timestamp, user_ID, user_group, prompt_num, prompt, response.

Preliminary Survey

A. Demographics & Background

Below are the questions included in the preliminary survey instrument.

Q.1 Name (first last):

Q.2 Email:

Q.3 Age:

Q.4 Ethnicity:

Q.5 Year:

- (a) Freshman
- (b) Sophomore
- (c) Junior
- (d) Senior
- (e) Grad student
- (f) Other:

Q.6 Gender:

Q.7 Major/minor/field of study:

Q.8 Category of field of study:

Q.9 Are you a native English speaker?

- (a) Yes
- (b) No

Q.10 Which generative AI tools have you used before? (Select all that apply)

- (a) ChatGPT
- (b) Bard
- (c) Claude
- (d) Co-pilot
- (e) Built-in tools (Grammarly, Word, etc.)
- (f) Other:

Q.11 How often do you currently use AI tools for writing tasks?

(1=Never) 1 2 3 4 5 (5=Always)

B. Self-Assessments

Rate your agreement with the following statements
(1 = Strongly disagree, 5 = Strongly agree).

Q.12 I consider myself a strong writer. ()

Q.13 I think AI will improve my writing ability. ()

Q.14 I worry that using AI will make me less capable as a writer. ()

Q.15 I am comfortable using AI tools for drafting, revising, or editing. ()

Q.16 I understand how to prompt AI (I know how to give clear prompts). ()

Q.17 Of your last 10 course-writing tasks, how many used AI in any way? (0–10):

C. Behavioral Baseline & Motivations

Q.18 Why do you use or avoid AI tools for writing?
(1–2 sentence open text)

Q.19 Any comments, questions, concerns?

Post-Interaction Survey

A. Usage Log (Only asked to AI groups)

Below are the questions included in the post-interaction survey instrument.

Q.1 Approximately what percent of your final text was directly generated or suggested by AI?

- (a) 0%
- (b) 1–10%
- (c) 11–25%
- (d) 26–50%
- (e) 51–75%
- (f) 76–99%
- (g) 100%

Q.2 Did you copy-paste any AI output into your final submission?

- (a) Yes
- (b) No

Q.3 Did you edit the AI output?

- (a) No edits
- (b) Minor edits (wording)
- (c) Moderate edits (structure)
- (d) Major edits (rewrote most)

B. Process & Experience

Rate your agreement with the following statements (1 = Strongly disagree, 5 = Strongly agree).

Q.4 The AI suggestions helped me generate ideas. ()

Q.5 The AI suggestions saved me time. ()

Q.6 The AI suggestions improved the clarity of my writing. ()

Q.7 The AI suggestions reduced my creative expression. ()

Q.8 I felt I relied too heavily on AI to complete the task. ()

Q.9 I had to do substantial critical thinking to decide what to accept or reject from the AI. . . ()

Q.10 I felt the final product still reflected my own voice. ()

Q.11 I would submit the final essay as my own in a regular class.

- (a) Yes
- (b) No
- (c) Not sure

Q.12 I would disclose to an instructor that I used AI for this piece.

- (a) Yes
- (b) No
- (c) Maybe

C. Cognitive Load / Effort & Time

Q.13 How difficult was it to complete this task?
(Very easy) 1 2 3 4 5 (Very difficult)

Q.14 How much of the 30 minutes did you spend editing vs. composing vs. reading AI outputs?
Editing: %
Composing: %
Reading AI outputs: %

D. Reflection on Skills

After this session, indicate how you feel your ability changed in each area (increase / no change / decrease).

Q.15 Organize ideas:

Q.16 Use precise vocabulary:

Q.17 Write creatively:

Q.18 Perform critical analysis:

E. Originality & Authenticity

Q.19 How original do you feel your final essay is?
(Not at all original) 1 2 3 4 5
(Very original)

Q.20 On a scale of 0–100, how authentic does the essay feel to you?

Q.21 Would you be comfortable if this exact essay were used as a graded assignment?

- (a) Yes
- (b) No
- (c) Maybe

F. Ethics & Academic Integrity

Q.22 Do you think using AI in coursework should be allowed?

- (a) Yes
- (b) No
- (c) Depends — please explain:

Q.23 If allowed, should students be required to disclose AI usage?

- (a) Yes
- (b) No
- (c) Depends

G. Open Reflection

Q.24 Describe (in 1–3 sentences) how you used AI in this task (what prompts you gave, what you accepted/rejected).

Q.25 What, if anything, did AI prevent you from doing or thinking about?

Q.26 Did anything about the AI outputs surprise you?

- (a) Yes — please explain:
- (b) No

Q.27 Any additional comments about how AI influenced your thinking or writing on this task?

H. Optional Detection / Guessing Task

Q.28 We sometimes run automatic classifiers on texts. Do you think your submitted essay was more likely:

- (a) Human-written
- (b) AI-assisted
- (c) Mostly AI-generated

Essay Grading Rubric

The following rubric was used to evaluate the quality of student essays across all experimental conditions. Each category is scored on a scale of 1–5 points.

Scoring: Each category is scored from 1–5 points. Total possible score for the first six categories: 30 points.

Category	5	4	3	2	1
Focus & Details	Engaging and full development of a clear thesis as appropriate to the assignment purpose.	Competent and well-developed thesis; thesis represents a sound and adequate understanding of the assigned topic.	Mostly intelligible ideas; thesis is weak, unclear, too broad, or only indirectly supported.	Most simplistic and unfocused ideas; little or no sense of purpose.	Ideas are extremely simplistic, showing signs of confusion and misunderstanding of the prompt. The thesis is missing or not discernible.
Ideas, Support, & Development (Evidence)	Consistent evidence with originality and depth of ideas. Ideas work together, and the main points are sufficiently supported with evidence.	Ideas are supported sufficiently; support is sound, valid, and logical.	Main points and ideas are only indirectly supported. Support is loosely relevant to the main points.	Insufficient or non-specific support.	Lack of support for main points.
Organization	Organization is sequential and appropriate. Paragraphs are well developed and appropriately divided.	Competent organization, without sophistication. May lack effective transitions.	Limited attempts to organize around a thesis; paragraphs are mostly stand-alones with weak transitions.	Organization, while attempted, was unsuccessful. Paragraphs were simple, disconnected, and formulaic.	Little to no organization; disjointed or confusing structure.
Voice	The author's purpose is very clear. Their extensive knowledge and/or experience with the topic is/are evident.	The author's purpose is mostly clear. Some evidence of attention to the audience. The author's knowledge and/or experience with the topic is/are evident.	Little or inconsistent sense of the audience related to the assignment purpose. Tone and point of view are not refined or consistent.	Shows almost no awareness of a particular audience; reveals no grasp of appropriate tone and/or point of view for given assignment.	Lacks awareness of a particular audience for the assignment; tone and point of view are somewhat inappropriate or very inconsistent.
Vocabulary and Word Choice	Exceptional vocabulary range, accuracy, and correct and effective word usage.	Good vocabulary range and accuracy of usage.	Ordinary vocabulary range, mostly accurate.	Errors of diction and usage, while evident, do not interfere with readability.	Extremely limited vocabulary; choices lack grasp of diction.
Sentence Structure, Grammar, Mechanics, & Spelling	All sentences are well constructed and have varied structure and length. The author makes no errors in grammar, mechanics, and/or spelling.	Most sentences are well constructed and have varied structure and length. Contains only occasional punctuation, spelling, and/or capitalization errors.	Sentences show a formulaic or tedious sentence pattern. Some errors in grammar, mechanics, and/or spelling, but they do not interfere with understanding.	Most sentences are well constructed, but they have a similar structure and/or length. The author makes several errors in grammar, mechanics, and/or spelling that interfere with understanding.	Sentences sound distractingly repetitive or are difficult to understand. The author makes numerous errors in grammar, mechanics, and/or spelling that interfere with understanding.
AI Detection/ Human-written	I am confident that this text was completely human-written. This essay feels completely authentic and original.	I am confident that this text was written mostly by a human. This essay feels authentic, but may have had some AI assistance.	I believe this essay was done by a human with AI assistance. Some portions of text seem to be AI-generated.	I believe this essay was mostly written with the help of AI. A large portion of the text sounds repetitive and/or unoriginal.	I believe this essay was completely written by AI.

Assessment of L2 Speech Global Dimensions using Large Audio Language Models

Elsayed Issa
Purdue University
West Lafayette, IN, USA
esissa@purdue.edu

Mahmoud Ali
University of Michigan
Ann Arbor, MI, USA
alimahm@umich.edu

Abstract

Large audio language models (LALMs) integrate audio representations with large language models to enable unified understanding of spoken content. Their capabilities have been increasingly investigated across several benchmarks; however, the examination of their use in rating L2 speech is still in its infancy. This study explores the abilities of LALMs in scoring three L2 speech global dimensions: foreign accentedness, comprehensibility, and intelligibility. Ninety audio samples produced by L2 speakers were rated by ten native speaker raters as well as five LALM models. Model performance was evaluated against the human composite mean using Pearson r , Spearman p , mean absolute error (MAE), and systematic bias, with the human leave-one-out correlation ($r = .46-.73$ across dimensions) serving as an empirical performance benchmark. The results showed that no LALM reached human-level performance on any dimension. Only one model (i.e., Gemini) achieved a significant correlation with human ratings on comprehensibility ($r = .28, p < .01$), while Qwen2-Audio showed modest correlation on intelligibility ($r = .32, p < .01$). MAE ranged from 0.75 to 3.99 for accentedness (human: 1.24), 1.35 to 3.00 for comprehensibility (human: 1.24), and 12.03 to 15.43 for intelligibility (human: 8.49). All models exhibited systematic biases, with deviations ranging from -9.31 to +13.19 points. The paper concludes with a discussion of the implications for automated L2 speech assessment.

1 Introduction

Large audio language models (LALMs) integrate audio representations with large language models to enable unified understanding and interaction over spoken content, environmental sounds, and even music. Such integration pushes beyond traditional automatic speech recognition pipelines toward more general instruction following and audio

question answering (Chu et al., 2024; Ding et al., 2025; Yang et al., 2024a; Tang et al., 2023). Recent models such as Qwen-Audio, Qwen2-Audio, Qwen2-Audio-7B-Instruct and Kimi-Audio demonstrate strong zero- and few-shot behavior across diverse audio tasks. These capabilities highlight how scale and multitask training can broaden audio reasoning and conversational capabilities (Chu et al., 2023, 2024; Ding et al., 2025).

As LALMs move from offline speech-to-text toward interactive voice assistants, the relevant notion of *understanding* increasingly encompasses not only speech but also non-speech audio, music, and other acoustic signals. These capabilities have been systematically investigated across a growing number of benchmarks, including AudioBench (Wang et al., 2025a), VoiceBench (Chen et al., 2024), SD-Eval (Ao et al., 2024), ADU-Bench (Gao et al., 2025), and MMAU (Wang et al., 2025b), each targeting different aspects of audio comprehension such as instruction following, spoken dialogue understanding, and multi-task audio reasoning. Previous work on L2 speech investigated GPT-4o capabilities for pronunciation assessment across phoneme, word, and sentence levels (Ahn and Nam, 2025), the potential of LALMs in language learning (Liu et al., 2026), multimodal LLM-based pronunciation scoring system (Fu et al., 2024), speech intelligibility (Pham, 2023), and ASR bias in relation to intelligibility, comprehensibility, and foreign accentedness (Issa et al., 2026). However, the investigation of L2 speech global dimension (i.e., intelligibility, comprehensibility, and foreign accentedness) is limited.

To address this issue, the current study aims to examine the extent to which LALMs can reliably and validly assess L2 speech listener-based global dimensions, namely intelligibility, comprehensibility, and foreign accentedness. These three constructs occupy a central place in L2 speech research (Munro and Derwing, 2015), yet the empirical in-

investigation of LALMs effectiveness in measuring these dimensions remains unknown. The study is guided by the following research questions: 1) How foreign accented, comprehensible, and intelligible is L2 Arabic speech based on LALMs assessment? 2) To what extent do LALMs agree with human rating of intelligibility, comprehensibility, and foreign accentedness in L2 Arabic speech? 3) How LALMs-based ratings of intelligibility, comprehensibility, and foreign accentedness differ as a function of model category?

2 Related work

2.1 L2 speech global dimensions

In L2 pronunciation research, a shift has moved the focus from native-like pronunciation toward comprehensible and intelligible speech (Levis, 2005; Nagle et al., 2018). In line with this shift, recent research has centered on three global, perceptual dimensions of L2 speech: foreign-accentedness (deviation from native-speaker norms), comprehensibility (ease of understanding), and intelligibility (actual understanding) (Munro and Derwing, 1995). These constructs have been found to be distinct yet partially related, with the key finding being that foreign-accented speech is not necessarily less comprehensible or intelligible across multiple languages (e.g., Huensch and Nagle, 2021; Munro and Derwing, 1995; Ali, 2023).

These three constructs have been examined in the context of L2 pronunciation research based on trained or untrained human raters. Munro and Derwing (1995) examined L2 English speech and reported significant correlations between comprehensibility and intelligibility for 15 of their 18 listeners, with a mean correlation of $r = .51$, while only five listeners showed significant correlations between accentedness and intelligibility. In L2 Spanish, Huensch and Nagle (2021) found that comprehensibility is strongly linked to intelligibility, but foreign-accentedness is not, and that phonemic errors have a stronger negative association with listener ratings than grammatical errors. Ali (2023) extended the investigation to L2 Arabic by examining the relationships among foreign-accentedness, comprehensibility, and intelligibility and exploring the contribution of foreign-accentedness and/or comprehensibility ratings in intelligibility scores. The results supported the partial independence of the three dimensions. Recent meta-analytic evidence has provided a comprehensive synthesis of

these dimensions relationships. Chau and Huensch (2025) analyzed 141 effect sizes from 49 studies spanning 1995 to 2023, documenting weighted mean correlations of $r = .75$ for foreign accentedness/comprehensibility, $r = .57$ for comprehensibility/intelligibility, and $r = .32$ for intelligibility/foreign accentedness. These findings establish a clear hierarchy in which intelligibility shows the weakest relationship with foreign accentedness and moderate associations with comprehensibility, while comprehensibility and foreign accentedness show stronger interdependence. The small correlation between intelligibility and foreign accentedness ($r = .32$, 95% CI [.08, .52]) provides the most compelling evidence that having a strong foreign accent does not necessarily impede actual understanding. Across studies in this line of research, human raters have been found to provide reliable and valid assessment of the three dimensions under investigation.

Efforts to automate the assessment of L2 speech have been examined, though most work has focused on specific speech dimensions such as segmental and suprasegmental features. Automatic speech scoring systems operationalize pronunciation differently and have been criticized for lack of transparency in how speech features are weighted (Cai et al., 2025). Traditionally, these systems rely on training procedures using neural networks. In addition, one of the limitations in these systems is the high error rates, particularly spontaneous L2 speech (Cai et al., 2025).

More recently, attention has turned to whether large language models and multimodal architectures can address this limitation. Ahn and Nam (2025) investigated the zero-shot capabilities of GPT-4o for pronunciation assessment across phoneme, word, and sentence levels. The study found that its scoring accuracy was significantly lower than tools that are specifically designed for that purpose, leading to the conclusion that multimodal LLMs may not yet replace purpose-built systems without fine-tuning. Similarly, Fu et al. (2024) proposed a multimodal LLM-based pronunciation scoring system and demonstrated competitive results against alignment-based baselines on the Speechocean762 dataset, though performance on spontaneous L2 speech and global dimensions remains unexamined.

Moreover, Pham (2023) investigated L2 English intelligibility using both the ASR transcription system (i.e., Otter) and 40 L2 English human

raters. The study found that while the ASR system achieved 79-91% transcription accuracy, human intelligibility scores were lower ($M = 3.9$ - 6.8 out of 10), revealing a critical divergence between ASR capabilities and human accuracy. In the same vein, Issa et al. (2026) examined linguistic bias in Whisper large-v3 performance on L2 Arabic speech, comparing ASR word error rate (WER) to human-based transcriptions (i.e., intelligibility) and ratings of comprehensibility and foreign accentedness. The findings revealed that intelligibility scores significantly predicted lower WER while foreign accentedness was associated with higher WER, whereas comprehensibility appear to be independent of WER. Finally, Liu et al. (2026) explored the potential of LALMs for interactive language learning by introducing L2-Arctic-plus, an English dataset annotated with detailed pronunciation error explanations and actionable suggestions for improvement. The study benchmarked both cascaded ASR-plus-LLM pipelines and existing LALMs on tasks including error detection and the generation of corrective feedback. Findings indicated that current models show promise for chat-based pronunciation training but still fall short of providing consistently accurate and pedagogically useful guidance.

2.2 LALMs benchmarks

A growing body of research has systematically evaluated LALM performance across diverse benchmarks. AudioBench (Wang et al., 2025a) provided a comprehensive evaluation framework spanning 8 task categories and 26 datasets while also examining prompt sensitivity effects in LALMs, where they found that LALMs could behave differently to different prompt templates. Across multiple audio modalities including speech, environmental sounds, and music, AIR-Bench (Yang et al., 2024b) assessed instruction-following competencies through a two-tier architecture: a foundation tier comprising 19 tasks with approximately 19,000 single-choice items, and a conversational tier containing roughly 2,000 open-ended question-answer pairs evaluated via GPT-4-as-judge methodology against reference responses. Likewise, MMAU (Tyagi et al., 2024) targeted audio understanding and reasoning across three domains (speech, sound, music) using exclusively multiple-choice formats to enable fully automated assessment without human annotation or LLM-based judging. Moreover, Dynamic-SUPERB (Huang et al., 2024b,a) imple-

ments a two-phase evaluation protocol for universal instruction-based speech models under zero-shot conditions. Results revealed that no single model achieves consistent performance across all tasks: SALMONN-13B demonstrates superiority in English ASR, whereas Qwen2-Audio-7B-Instruct exhibits stronger emotion recognition capabilities, yet neither generalizes effectively across the complete task spectrum.

Several benchmarks have targeted contextual and content-level comprehension. SD-Eval (Ao et al., 2024), for instance, addresses a more focused yet critical aspect of LALM functionality: spoken dialogue comprehension beyond surface-level transcription. It assesses models' capacity to simultaneously process content, speaking style, speaker emotion, and background acoustics—the multifaceted contextual understanding required for conversational assistants. Findings indicated inadequate contextual response generation, attributed to insufficient task formalization and appropriate evaluation datasets. Similarly, ADU-Bench (Gao et al., 2025) evaluates open-ended audio dialogue performance across 3 general scenarios, 12 competencies, 9 languages, and 4 ambiguity categories. Evaluation of 16 LALMs revealed superior performance on information-seeking queries relative to casual conversational exchanges, with systematic failures in mathematical reasoning, roleplay interactions, non-English language processing, and phonetic ambiguity resolution.

Additional benchmarks have specifically investigated audio hallucination phenomena. Ghosh et al. (2023) introduced Comp-A to assess compositional reasoning, demonstrating that contemporary models achieve only marginally above-chance performance when matching audio clips to captions with varied event orderings or attribute bindings. Sung-Bin et al. (2024) developed AVH-Bench, a cross-modal hallucination benchmark for audio-visual LLMs, revealing widespread difficulty in parsing fine-grained audio-visual relationships. Furthermore, AHa-Bench (Cheng et al., 2025) proposed a tripartite hallucination taxonomy—semantic, acoustic, and semantic-acoustic—and evaluated seven open-source LALMs, uncovering substantial limitations in models' joint interpretation of semantic and acoustic information. While these benchmarks have examined several capabilities of LALMs, they mainly focused on L1 English and L1 Chinese, leaving not only other L1s but also L2s unexplored, including Arabic.

3 Methods

3.1 Data

The L2 Arabic speech data came from 30 adult L1 English speakers (19M, 11F) who were enrolled in second- and third-year Arabic courses at a U.S. public university. All were rated at the Intermediate level of proficiency based on the American Council on the Teaching of Foreign Languages (ACTFL) scale. The speech samples were rated by 10 L1 Arabic speakers (9M, 1F) with minimal exposure to L2 Arabic speech and no teaching experience. No participants reported hearing or speech impediments (Ali, 2023).

Speech Production: The speech samples were as part of an Oral Proficiency Interview (OPI) administered by a certified OPI tester and recorded via Zoom (15–35 min) following an online language background questionnaire. Three speech samples per speaker were extracted from distinct OPI sections. Samples were amplitude-normalized (peak = 0.99) in Audacity (Audacity Team, 2020), yielding 90 samples (30 speakers x 3) averaging 13 seconds and 15 words.

Speech Perception: Ten L1 Arabic listeners completed three tasks on all 90 samples across counter-balanced sessions on separate days (3 hours total). Task 1 involved rating foreign-accentedness on 9-point scale where 0 means no foreign accent and 9 means strongly foreign accented. Task 2 involved rating comprehensibility on 9-point scale where 0 means very easy to understand and 9 means very difficult to understand. Finally, in task 3, listeners transcribed the speech samples as a measure of intelligibility (Munro and Derwing, 1995). An intelligibility score was calculated as the percentage of correctly transcribed words. Tasks were blocked (35 samples/block) and presented via a custom Qualtrics interface, with listeners rating each sample after a single hearing following three practice items.

3.2 Experiments

Five LALMs were purposively selected due to their multilingual capabilities, including Arabic. These models are Qwen2-Audio¹, Qwen2-Audio-7B-Instruct² (Chu et al., 2024), Ultravox³, Gemini-

2.5-Pro (Google DeepMind, 2025) and GPT-4o-audio-preview (OpenAI, 2024) (see Table 4 for more information about the open-source models). We probed each model using a standardized zero-shot prompt designed to elicit numerical ratings on each of the three global speech dimensions. Models are prompted independently per dimension using the prompts shown in Table 5. To evaluate the extent to which these five LALMs can simulate human perception of L2 speech dimensions under investigation, a series of statistical analyses were conducted in Python using the pingouin library (Vallat, 2018).

Our analyses map onto the three research questions as follows. For RQ1, descriptive statistics (mean, SD, range, 95% CI) summarize how LALMs rate each dimension. For RQ2, we evaluate LALM–human agreement using complementary metrics that capture distinct facets of agreement: Pearson r and Spearman ρ index linear and rank-order correspondence; mean absolute error (MAE) and root mean square error (RMSE) quantify the magnitude of per-item disagreement; and signed bias captures systematic over- or underestimation. We additionally fit each LALM as an 11th rater alongside the ten human raters and recompute the intraclass correlation, testing whether a LALM behaves as an exchangeable member of the rater panel. A leave-one-out (LOO) analysis, in which each human rater’s ratings are correlated with the mean of the remaining nine, provides an empirically grounded benchmark for human-level performance against which LALM agreement can be interpreted.

For RQ3, we compare per-item MAE between open- and closed-source model families using a Wilcoxon signed-rank test (paired by item), with Cohen’s d as a measure of effect size. The Wilcoxon test was chosen over a paired t-test because per-item MAE is bounded at zero and skewed, violating the normality assumption. We measure Intraclass Correlation Coefficient (ICC) variants (Shrout and Fleiss, 1979; McGraw and Wong, 1996). On the one hand, ICC2 is the single-rater absolute-agreement coefficient. It indicates how closely one randomly selected rater (human or LALM) matches another in absolute terms. ICC2k, on the other hand, is the corresponding average-measures coefficient, showing the reliability of the mean rating across the full panel. ICC3 and ICC3k are the consistency analogs (two-way mixed effects), which discount systematic between-rater

¹<https://huggingface.co/Qwen/Qwen2-Audio-7B>

²<https://huggingface.co/Qwen/Qwen2-Audio-7B-Instruct>

³https://huggingface.co/fixie-ai/ultravox-v0_5-llama-3_1-8b

shifts in mean; we report these for completeness. ICC2 is our primary metric because the research question is whether a LALM produces the same numerical rating a human would, not merely a rank-consistent one. This same logic motivates reporting MAE and signed bias alongside Pearson r : Pearson correlation is invariant to additive shifts that absolute-agreement metrics correctly penalize.

The five LALMs differ systematically along a dimension that is theoretically relevant to L2 Arabic assessment: training scale and Arabic representation. The closed-source models (Gemini 2.5-Pro, GPT-4o-audio-preview) are trained on substantially larger and more multilingual corpora than the open-source models in our sample, which are built on Whisper-large-v2 or v3-turbo encoders paired with 7–8B parameter language model backbones (Qwen-7B, Llama-3.1-8B; see Table 4). Arabic, and L2 Arabic in particular, is plausibly underrepresented in the training data of the open-source models. Prior benchmark work has shown that LALMs struggle in non-English language environments (Gao et al., 2025), making the open- versus closed-source comparison a useful probe of whether scale and broader multilingual coverage partially compensate for the demands of low-resource L2 perceptual rating. We therefore examine whether per-item agreement with human raters differs systematically between the two model families, recognizing that with five models this comparison is hypothesis-generating rather than definitive.

4 Results

Table 1 presents the descriptive statistics for LALMs ratings across the three dimensions. LALMs rated foreign accentedness toward the higher end of the 9-point scale ($M = 5.70$, $SD = 2.40$), comprehensibility ratings fell near the midpoint of the scale ($M = 4.78$, $SD = 2.47$), and intelligibility showed the broadest range among the three measures ($M = 77.08$, $SD = 28.85$, range = 20–100) (see Figure 3 for distribution of comprehensibility and foreign-accentedness ratings and intelligibility scores by human raters and LALMs).

Table 2 presents Pearson r , mean absolute error (MAE), and systematic bias for each model across the three dimensions, alongside the human LOO analysis. Across all dimensions and all five models, no LALM reached the human LOO correlation threshold. All models showed statistically significant deviations from the human agreement

composite mean (Wilcoxon signed-rank, all $p < .001$; see Table 7). For intelligibility, Qwen2-Audio was the only model to significantly correlate with human ratings ($r = .322$, $p = .002$). On the other hand, Gemini and GPT showed no significant correlations.

Ultravox and Qwen2-Audio-Instruct provided the same scores/ratings to all speech samples across all the three dimensions. From statistical analysis perspective, we acknowledge that these are considered statistical artifacts. However, from model performance evaluation, this is also indicative of model hallucinations which will be discussed later.

As for comprehensibility, while the association between Gemini and human agreement composite mean was significant ($r = .281$, $p = .007$), the correlation between GPT and the human agreement composite mean was not ($r = .180$, $p = .090$). Open-source models did not correlate meaningfully. Finally, for foreign accentedness, none of the LALMs significantly correlated with human-based ratings. It is worth noting that Ultravox and Qwen2-Audio-Instruct rated speech samples with constant scores across constructs, resulting in meaningless correlations with those of humans.

Table 3 summarizes the results that examine whether model category influenced model performance. a Wilcoxon signed-rank test was adopted for each dimension comparing per-item mean absolute error between the open- and closed-source families. Per-item error of open-sourced models was significantly larger than that of closed-sourced models for intelligibility (open MAE = 29.02, $SD = 5.02$; closed MAE = 13.76, $SD = 10.80$) and for comprehensibility (open MAE = 2.49, $SD = 0.57$; closed MAE = 2.10, $SD = 1.05$) while there was no significance difference between the two model categories for foreign accentedness (open MAE = 2.07, $SD = 0.42$; closed MAE = 2.10, $SD = 0.87$).

Finally, we examined agreement reliability model fit with human-based agreement as baseline. Each LALM was fitted as an additional (11th) rater, and ICC2 was recomputed for the augmented panel and compared against the ten-human baseline. As shown in Figure 1, across all three dimensions, model fitting declined by adding any of the LALMs. For intelligibility, the decline was most pronounced for Gemini (baseline ICC2 = .49; with Gemini = .39) followed with GPT (baseline ICC2 = .49; with GPT = .43), and finally Qwen2-Audio (baseline ICC2 = .49; with Qwen2 = .44). As for comprehensibility, the decline was shown for Qwen2-Audio

	M	SD	Min	Max	95% CI
Foreign-accentedness	5.70	2.40	1	9	5.47-5.92
Comprehensibility	4.78	2.47	1	9	4.55-5.00
Intelligibility	77.08	28.85	20	100	74.40-79.75

Table 1: Descriptive statistics for LALMs’ ratings of Foreign-accentedness, Comprehensibility, and Intelligibility

		Foreign Accentedness			Comprehensibility			Intelligibility		
		<i>r</i>	MAE	Bias	<i>r</i>	MAE	Bias	<i>r</i>	MAE	Bias
Open	Qwen2-Audio	.15	3.99	3.90	.16	3.00	1.32	.32**	12.03	+8.80
	Qwen2-Instruct	—†	0.75	+0.48	—†	1.35	+1.11	—†	12.28	9.31
	Ultravox	—†	1.48	+1.48	—†	1.35	+1.11	—†	12.28	9.31
Closed	Gemini	.15	1.56	+0.39	.28**	2.11	1.40	.12	15.43	+13.19
	GPT	.04	2.64	2.58	.18	2.09	2.06	.07	12.08	+10.02
	Human LOO	.46	1.24	—	.57	1.24	—	.73	8.49	—

MAE = mean absolute error. Bias = model mean minus human mean. Human LOO = leave-one-out benchmark: mean Pearson *r* and MAE of each human rater correlated against the remaining nine raters, averaged across raters. † Zero-variance output where correlation is undefined. ** $p < .01$. See Table 9 for individual rater LOO correlations and MAE across dimensions

Table 2: Per-model agreement with human ratings across the three constructs (N = 90).

(baseline ICC2 = .336; with Qwen2 = .219), GPT (baseline ICC2 = .336; with GPT = .27), and Gemini (baseline ICC2 = .336; with Gemini = .29). Finally, the decline is pronounced for Qwen2-Audio (baseline ICC2 = .209; with Qwen2 = .122), followed by GPT (.15) and Gemini (.18) respectively for foreign-accentedness (see Table 8 for more on model fit).

5 Discussion

5.1 LALMs assessment of global speech dimensions

LALMs rated L2 Arabic speech as moderately to strongly foreign accented (M = 5.70, SD = 2.40), moderately difficult to understand (M = 4.78, SD = 2.47), and mostly highly intelligible (M = 77.08, SD = 28.85). However, aggregate similarity in mean ratings is not, by itself, evidence that LALMs reproduce human rating behavior. The within-dataset evidence presented in Sections 5.2 and 5.3, item-level agreement, distributional patterns, and the effect of substituting LALMs into the rater panel, yields a less favorable assessment. Prior work has documented limitations of multimodal LLMs in segmental and sentence-level pronunciation assessment (Ahn and Nam, 2025; Fu et al., 2024); the present results extend the scope of these limitations to global perceptual dimensions, where the gap between LALM and human raters, quantified against the human LOO benchmark, is more pronounced.

5.2 Global speech dimensions: LALMs-based versus human-based

Three sources of within-dataset evidence converge on the conclusion that LALMs do not reliably reproduce human rating behavior on our corpus.

Item-level agreement: No LALM reached the human LOO correlation benchmark on any dimension (Table 3). The human LOO correlations averaged $r = .73$ for intelligibility, .57 for comprehensibility, and .46 for foreign accentedness, whereas the strongest LALM correlations were $r = .32$ (Qwen2-Audio, intelligibility), .28 (Gemini, comprehensibility), and .15 (Qwen2-Audio and Gemini, foreign accentedness). Per-item MAE values likewise exceeded the human LOO MAE on every dimension and for every model. Even the closest match, Qwen2-Instruct on foreign accentedness (MAE = 0.75), is misleading, as that value reflects a constant output rather than item-sensitive rating.

Distributional patterns: Although LALM mean ratings approximate human means at the aggregate level, Figure 2 shows that the underlying distributions diverge. LALM ratings exhibit larger standard deviations and, in two cases (Ultravox and Qwen2-Instruct), collapse to a single value across all 90 samples. This distributional mismatch indicates that aggregate similarity in central tendency masks substantial item-level disagreement.

Effect on inter-rater reliability: When each LALM is fitted as an 11th rater alongside the

Construct	Open MAE	Closed MAE	W	<i>p</i>	<i>d</i>
Foreign Accentedness	2.07	2.10	1934.5	.781	0.03
Comprehensibility	2.49	2.10	1355.0	.008	0.27
Intelligibility	29.02	13.76	263.0	< .001	1.11

Table 3: Open-source vs. closed-source comparison of per-item mean absolute error across the three constructs. *W* and *p* are from Wilcoxon signed-rank tests (paired by item); *d* is Cohen’s *d*. The Wilcoxon test was chosen because per-item MAE is bounded at zero and right-skewed, violating the normality assumption of the paired t-test.

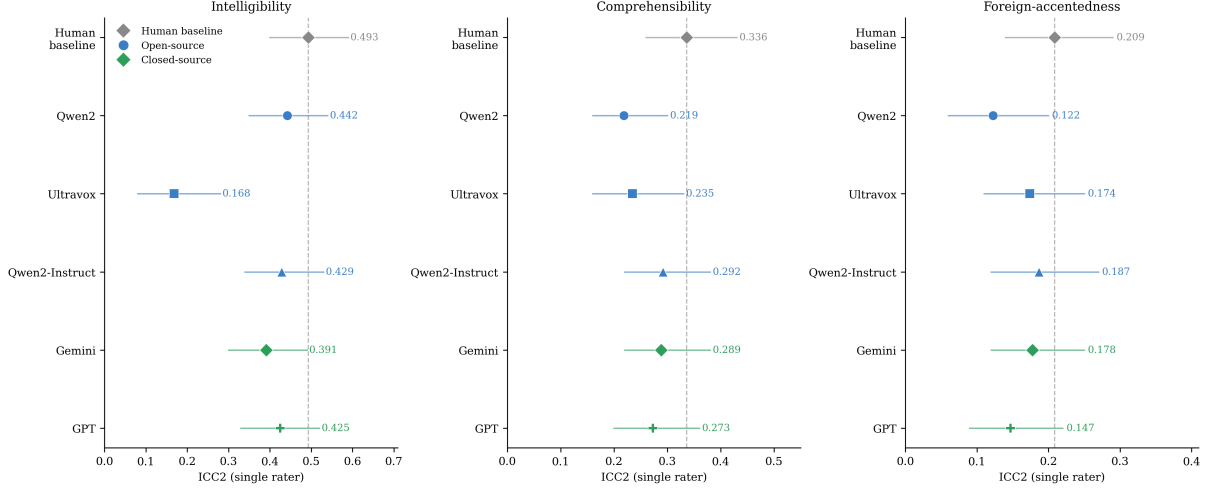


Figure 1: ICC2 when each model is added as an 11th rater to the human panel (dashed line = 10-human baseline; error bars = 95% CI)

ten human raters, the panel’s single-rater absolute-agreement coefficient (ICC2) declines on every dimension and for every model (Figure 1; Table 8). The decline is most pronounced for intelligibility with Gemini (.49 → .39), for comprehensibility with Qwen2-Audio (.34 → .22), and for foreign accentedness with Qwen2-Audio (.21 → .12). Adding any LALM to the rater panel reduces, rather than augments, the reliability of the rating panel which is a direct evidence that LALMs do not function as exchangeable raters.

Taken together, these three lines of evidence indicate that the apparent similarity between LALM and human ratings at the aggregate level does not extend to the item level, the distributional level, or the level of rater interchangeability. This pattern is consistent with prior findings of divergence between automated and human speech assessment (Pham, 2023) and suggests that LALMs do not simulate the underlying mechanisms human raters draw on when evaluating L2 speech.

5.3 Open- versus closed-model assessment

The by-category comparison reveals a dimension-dependent pattern that aligns with the theoretical motivation outlined in Section 3.2. Closed-source

models outperformed open-source models on intelligibility (open MAE = 29.02, closed MAE = 13.76; $d = 1.11$, $p < .001$), modestly outperformed them on comprehensibility (open MAE = 2.49, closed MAE = 2.10; $d = 0.27$, $p = .008$), and showed no advantage on foreign accentedness (open MAE = 2.07, closed MAE = 2.10; $d = 0.03$, $p = .781$). Figure 2 and Table 10 corroborate this gradient via Bland-Altman statistics. The dimension-dependent gradient is informative about what kind of capability drives performance on each construct. Intelligibility is operationalized as the proportion of words correctly transcribed, so it depends most directly on ASR quality, a capability that scales with training data volume and multilingual coverage, where closed-source models have a clear advantage. Comprehensibility is a perceptual judgment of processing effort that is less directly tied to ASR quality, which may explain the smaller advantage. Foreign accentedness requires phonetic comparison against L1 Arabic norms, a capability that appears to be poorly developed in both model families and is consistent with Arabic being underrepresented in training data across the board. Read this way, the open/closed comparison is not

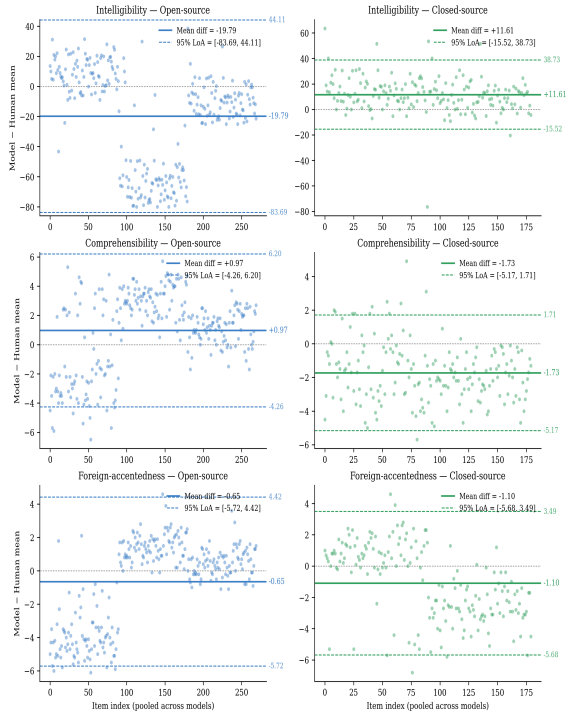


Figure 2: Model category ratings vs. human composite mean

merely a performance comparison between model families but evidence that scale and multilingual coverage partially compensate for low-resource L2 settings on transcription-driven dimensions, while leaving phonetic-norm-based judgments largely unaddressed.

The constant-output behavior observed in Ultra-vox and Qwen2-Audio-Instruct warrants attention. Both models produced identical ratings across all 90 samples on every dimension, which is reflected in their undefined Pearson correlations (Table 3) and in the spike-shaped distributions in Figure 3. This pattern indicates that the models did not engage with item-level acoustic or linguistic variation when prompted for perceptual ratings, defaulting instead to a fixed response. This behavior is reported as hallucination phenomena documented in LALM benchmarks (Cheng et al., 2025; Sung-Bin et al., 2024), in which models generate plausible but unfounded outputs. This raises the question of whether instruction-tuned LALMs are prone to substituting heuristic responses for genuine acoustic analysis on perceptual rating tasks. Possible explanations include insufficient exposure to L2 Arabic in pretraining/instruction-tuning that biases the model toward modal scale-point responses regardless of input, or failure of the audio encoder to extract dimension-relevant features from the signal.

5.4 Implications for automated L2 speech assessment

The findings of this study carry several implications for the use of LALMs in L2 speech assessment. First, current LALMs may not be used in high-stakes assessment contexts such as proficiency certification or placement testing. This aligns with the broader consensus in language assessment that automated scoring systems must accumulate validity evidence comparable to that expected of human raters before they can be justified for operational use, especially in high-stakes decisions (Chapelle and Voss, 2021). Second, the finding that certain models showed partial sensitivity to specific dimensions suggests that LALMs may be amenable to domain-specific fine-tuning for targeted dimension assessment. Rather than relying on general-purpose multitask training, future work could fine-tune LALMs on L2 speech data annotated for specific global dimensions, potentially improving model–human agreement on dimensions where zero-shot performance has the greatest potential. Finally, the results highlight the need for closer collaboration between the speech technology researchers and applied linguistics communities. Current LALM benchmarks (Wang et al., 2025a; Tyagi et al., 2024; Wang et al., 2025b; Sung-Bin et al., 2024) evaluate models primarily on L1 speech, but not L2 speech in their assessment criteria. One benchmark, (Gao et al., 2025), indicated that LALMs struggle in non-English language environments. Our findings demonstrate that strong performance on standard audio understanding tasks does not transfer to the perceptual demands of L2 speech rating. Future benchmark development should incorporate L2 speech assessment tasks across multiple languages and proficiency levels.

6 Conclusion

This study provided an evaluation of LALMs as automated raters across three global dimensions of L2 speech. Current LALMs cannot simulate human ratings with sufficient accuracy or reliability to serve as raters in L2 speech assessment. This conclusion rests on three converging within-dataset findings: LALM agreement with human ratings falls below the human LOO benchmark on every dimension; LALM rating distributions diverge from human distributions despite similar means; and adding any LALM to the rater panel reduces rather than augments panel reliability.

Limitations

Several limitations should be acknowledged. First, the sample of 90 recordings represents a relatively small corpus of L2 speech. We recognize this not only as a limitation but also as an invitation to L2 researchers to evaluate LALMs on their own datasets and to propose strategies for improving LALM performance. Second, we acknowledge that Arabic may be underrepresented in the training data of the open-source models examined in the current study. That said, this presents an opportunity to the broader L2 community to extend this line of inquiry to other languages in order to validate our findings. Third, models were evaluated in a zero-shot configuration using a single standardized prompt per dimension, without few-shot examples, chain-of-thought reasoning, et cetera. We chose this design deliberately to establish a baseline of out-of-the-box LALM behavior on L2 speech rating, which mirrors how non-expert end users (e.g., learners, instructors) are most likely to deploy these systems in practice. We acknowledge, however, that this design cannot disentangle inherent model limitations from prompt-induced underperformance, and some portion of the observed model-human gap may be attributable to suboptimal prompting rather than representational shortcomings. That said, two patterns in our results suggest prompting alone is unlikely to fully close the gap. First, the constant-output behavior of Ultravox and Qwen2-Audio-Instruct across all 90 samples points to a failure to engage with the acoustic signal rather than a calibration issue that anchors could resolve. Second, the magnitude of systematic bias observed for several models (up to +13.19 points on intelligibility) and the dimension-specific divergence between open- and closed-source families suggest underlying differences in audio representation quality, not merely scale-mapping artifacts. Future work should systematically vary prompting strategy to quantify the share of variance attributable to prompt design versus model capability.

Ethics Statement

We adhered to ethical principles to ensure responsible and respectful collection of the data. Participants were fully informed about the purpose of the study, the nature of the data being collected, and their right to withdraw at any time without consequence. No personally identifiable information was collected (see [Ali, 2023](#) for more information).

Data Availability

The L2 Arabic speech corpus and human ratings are not yet publicly available due to ongoing research, but the authors plan to release them in the near future. Access may be granted to researchers upon reasonable request.

Acknowledgment

We would like to thank the anonymous reviewers for their valuable comments, which helped to improve this manuscript; all remaining errors are our own. We also gratefully acknowledge the financial support and resources provided by the School of Languages and Cultures at Purdue University.

References

- Taekyung Ahn and Hosung Nam. 2025. English pronunciation evaluation without complex joint training: Lora fine-tuned speech multimodal llm. *arXiv preprint arXiv:2509.02915*.
- Mahmoud M. E. Ali. 2023. [The foreign-accentedness, comprehensibility, and intelligibility of l2 arabic speech](#). *Language Teaching Research*, 0(0).
- Junyi Ao, Yuancheng Wang, Xiaohai Tian, Dekun Chen, Jun Zhang, Lu Lu, Yuxuan Wang, Haizhou Li, and Zhizheng Wu. 2024. Sd-eval: A benchmark dataset for spoken dialogue understanding beyond words. *Advances in Neural Information Processing Systems*, 37:56898–56918.
- Audacity Team. 2020. [Audacity](#).
- Danwei Cai, Ben Naismith, Maria Kostromitina, Zhongwei Teng, Kevin P Yancey, and Geoffrey T LaFlair. 2025. Developing an automatic pronunciation scorer: Aligning speech evaluation models and applied linguistics constructs. *Language Learning*, 75(S1):170–203.
- Carol A Chapelle and Erik Voss. 2021. Introduction to validity argument in language testing and assessment. *Validity argument in language testing: Case studies of validation research*, pages 1–16.
- Tuc Chau and Amanda Huensch. 2025. The relationships among l2 fluency, intelligibility, comprehensibility, and accentedness: A meta-analysis. *Studies in Second Language Acquisition*, 47(1):282–307.
- Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T Tan, and Haizhou Li. 2024. Voicebench: Benchmarking llm-based voice assistants. *arXiv preprint arXiv:2410.17196*.
- Xize Cheng, Dongjie Fu, Chenyuhao Wen, Shannon Yu, Zehan Wang, Shengpeng Ji, Siddhant Arora, Tao Jin, Shinji Watanabe, and Zhou Zhao. 2025. Aha-bench: Benchmarking audio hallucinations in large

- audio-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.
- Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.
- Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, and 1 others. 2025. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*.
- Kaiqi Fu, Linkai Peng, Nan Yang, and Shuran Zhou. 2024. Pronunciation assessment with multi-modal large language models. *arXiv preprint arXiv:2407.09209*.
- Kuofeng Gao, Shu-Tao Xia, Ke Xu, Philip Torr, and Jindong Gu. 2025. Benchmarking open-ended audio dialogue understanding for large audio-language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4763–4784.
- Sreyan Ghosh, Ashish Seth, Sonal Kumar, Utkarsh Tyagi, Chandra Kiran Reddy Evuru, S Sakshi, Oriol Nieto, Ramani Duraiswami, Dinesh Manocha, and 1 others. 2023. Compa: Addressing the gap in compositional reasoning in audio-language models. In *The Twelfth International Conference on Learning Representations*.
- Google DeepMind. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). Technical report, Google DeepMind.
- Chien-yu Huang, Wei-Chih Chen, Shu-wen Yang, Andy T Liu, Chen-An Li, Yu-Xiang Lin, Wei-Cheng Tseng, Anuj Diwan, Yi-Jen Shih, Jiatong Shi, and 1 others. 2024a. Dynamic-superb phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks. In *The Thirteenth International Conference on Learning Representations*.
- Chien-yu Huang, Ke-Han Lu, Shih-Heng Wang, Chi-Yuan Hsiao, Chun-Yi Kuan, Haibin Wu, Siddhant Arora, Kai-Wei Chang, Jiatong Shi, Yifan Peng, and 1 others. 2024b. Dynamic-superb: Towards a dynamic, collaborative, and comprehensive instruction-tuning benchmark for speech. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12136–12140. IEEE.
- Amanda Huensch and Charlie Nagle. 2021. The effect of speaker proficiency on intelligibility, comprehensibility, and accentedness in l2 spanish: A conceptual replication and extension of munro and derwing (1995a). *Language Learning*, 71(3):626–668.
- Elsayed Issa, Mahmoud Ali, and Kevin Hirschi. 2026. [Measuring linguistic bias in ASR: Whisper large-v3 on non-native speech versus human perception](#). *Procedia Computer Science*, 275:692–699.
- John M Levis. 2005. Changing contexts and shifting paradigms in pronunciation teaching. In *Pronunciation*, pages 265–272. Routledge.
- Hongfu Liu, Zhouying Cui, Xiangming Gu, and Ye Wang. 2026. Unlocking large audio-language models for interactive language learning. *arXiv preprint arXiv:2601.14744*.
- Kenneth O. McGraw and S. P. Wong. 1996. [Forming inferences about some intraclass correlation coefficients](#). *Psychological Methods*, 1(1):30–46.
- Murray J Munro and Tracey M Derwing. 1995. Foreign accent, comprehensibility, and intelligibility in the speech of second language learners. *Language learning*, 45(1):73–97.
- Murray J Munro and Tracey M Derwing. 2015. A prospectus for pronunciation research in the 21st century: A point of view. *Journal of Second Language Pronunciation*, 1(1):11–42.
- Charles Nagle, Rebecca Sachs, and Germán Zárate-sánchez. 2018. Exploring the intersection between teachers’ beliefs and research findings in pronunciation instruction. *The Modern Language Journal*, 102(3):512–532.
- OpenAI. 2024. [GPT-4o System Card](#). Technical report.
- Trang Minh Thi Pham. 2023. The intelligibility of vietnamese-accented english to artificial intelligence software and asian listeners. Technical report, TESOL Working Paper Series.
- Patrick E. Shrout and Joseph L. Fleiss. 1979. [Intra-class correlations: Uses in assessing rater reliability](#). *Psychological Bulletin*, 86(2):420–428.
- Kim Sung-Bin, Oh Hyun-Bin, JungMok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun Oh. 2024. Avhbench: A cross-modal hallucination benchmark for audio-visual large language models. In *The Thirteenth International Conference on Learning Representations*.
- 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. In *The Twelfth International Conference on Learning Representations*.

Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Rameswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. 2024. Mmau: A massive multi-task audio understanding and reasoning benchmark. *arXiv preprint arXiv:2410.19168*.

Raphael Vallat. 2018. *Pingouin: statistics in python*. *Journal of Open Source Software*, 3(31):1026.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F. Chen. 2025a. *AudioBench: A universal benchmark for audio large language models*. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4297–4316, Albuquerque, New Mexico. Association for Computational Linguistics.

Dingdong Wang, Jincenzi Wu, Junan Li, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. 2025b. Mmsu: A massive multi-task spoken language understanding and reasoning benchmark. *arXiv preprint arXiv:2506.04779*.

Dongchao Yang, Haohan Guo, Yuanyuan Wang, Rongjie Huang, Xiang Li, Xu Tan, Xixin Wu, and Helen Meng. 2024a. Uniaudio 1.5: Large language model-driven audio codec is a few-shot audio task learner. *Advances in Neural Information Processing Systems*, 37:56802–56827.

Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, and Jingren Zhou. 2024b. *AIR-bench: Benchmarking large audio-language models via generative comprehension*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1979–1998, Bangkok, Thailand. Association for Computational Linguistics.

A Data Annotation Guidelines

The annotation procedure followed established guidelines from the L2 pronunciation research literature, specifically the protocol introduced by [Munro and Derwing \(1995\)](#) and subsequently adopted in numerous studies of L2 speech global dimensions (e.g., [Huensch and Nagle, 2021](#); [Chau and Huensch, 2025](#)). The guidelines operationalize each of the three constructs as follows: foreign accentedness was rated on a 9-point scale (1 = no foreign accent, 9 = strong foreign accent); comprehensibility was rated on a 9-point scale (1 = very easy to understand, 9 = very difficult to understand); and intelligibility was operationalized as the proportion of correctly transcribed words relative to the total number of words actually spoken (0–100).

The full operational definitions provided to raters are reported in Table 5 of the manuscript and are identical to the prompts used to elicit ratings from the LALMs, ensuring comparability between human and model ratings. The complete annotation protocol, including listener recruitment, task instructions, scale anchors, and session structure, is documented in ([Ali, 2023](#)).

B Model Information

Qwen2-Audio
Qwen2-Audio builds upon its predecessor Qwen-Audio (Chu et al., 2023), integrating audio and text inputs through a unified encoder-decoder architecture to generate textual outputs. The model processes audio inputs using a Whisper-large-v2 audio encoder paired with a Qwen-7B large language model backbone. It introduces two distinct audio interaction modes: audio analysis and voice chat.
Qwen2-Audio-Instruct
Qwen2-Audio-7B-Instruct is the instruction-tuned variant of Qwen2-Audio, built on the same Whisper-large-v2 encoder and Qwen-7B backbone architecture but further optimized for interactive dialogue through supervised fine-tuning on spoken dialogue data and instruction-following datasets.
Ultravox
Ultravox v0.5 is a multimodal speech language model that integrates a pretrained Llama-3.1-8B-Instruct backbone with the encoder component of Whisper-large-v3-turbo to process both speech and text inputs and generate textual outputs.

Table 4: Open-source model information

C Appendix

Appendix C reports supplementary materials referenced throughout the paper and the zero-shot prompts.

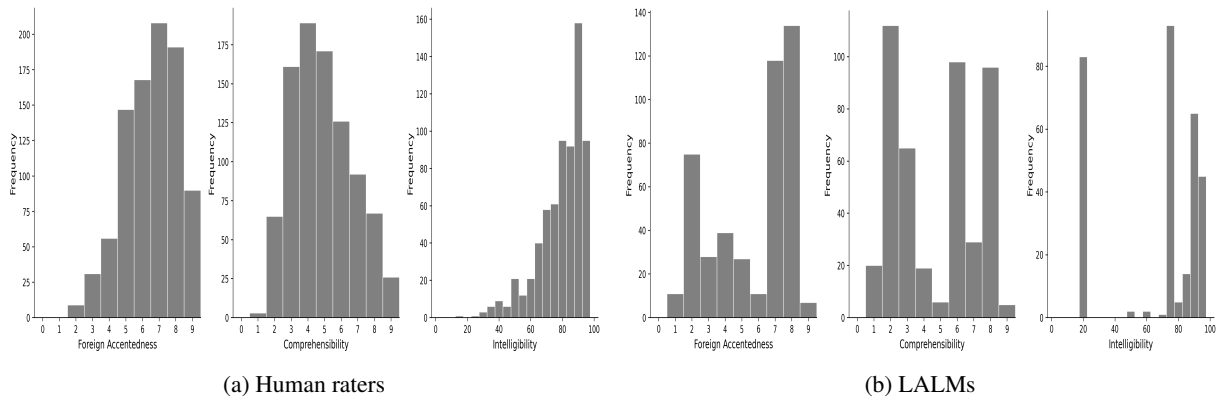


Figure 3: Distribution of of comprehensibility and foreign-accentedness ratings and intelligibility scores by human raters and LALMs.

Foreign Accentedness	
Definition:	The degree to which L2 speech is similar to or different from L1 speech in terms of pronunciation as perceived by a native L1 listener with no familiarity with L2 speech.
Operationalization:	9-point scale 1=Very similar to L1 speech in pronunciation (no foreign accent at all) 9-Very different to L1 speech (strong foreign accent)
Comprehensibility	
Definition:	The degree to which L2 speech is easy or difficult o understand as perceived by a native L1 listener with no familiarity with L2 speech.
Operationalization:	9-point scale 1=Very easy to understand (processing effort is zero to little) 9-Very difficult to understand (processing effort is maximum)
Intelligibility	
Definition:	The degree to which L2 speech is correctly recognized as transcribed by a native L1 listener with no familiarity with L2 speech.
Operationalization:	0-100 The proportion of correctly transcribed words as compared to the total number of words actually spoken.

Table 5: Zero-shot prompts for rating foreign accentedness, comprehensibility, and intelligibility.

ICC type	Foreign Accentedness			Comprehensibility			Intelligibility		
	ICC	95% CI	<i>p</i>	ICC	95% CI	<i>p</i>	ICC	95% CI	<i>p</i>
ICC2	.209	[.14, .29]	< .001	.336	[.26, .43]	< .001	.493	[.40, .59]	< .001
ICC2k	.725	[.62, .81]	< .001	.835	[.78, .88]	< .001	.907	[.87, .94]	< .001
ICC3	.725	[.62, .81]	< .001	.361	[.28, .45]	< .001	.549	[.47, .63]	< .001
ICC3k	.784	[.71, .84]	< .001	.850	[.80, .89]	< .001	.924	[.90, .95]	< .001

Table 6: Intraclass correlation coefficients for the ten human raters under a two-way random-effects model (Shrout and Fleiss, 1979). ICC2 = single-rater absolute agreement; ICC2k = average-rater absolute agreement (reliability of the panel mean); ICC3 = single-rater consistency (two-way mixed); ICC3k = average-rater consistency.

Model	Foreign Accentedness		Comprehensibility		Intelligibility	
	W	<i>p</i>	W	<i>p</i>	W	<i>p</i>
Qwen2	19.5	<.001	1144.0	<.001	508.0	<.001
Qwen2-Instruct	787.0	<.001	321.5	<.001	443.5	<.001
Ultravox	4.0	<.001	0.0	<.001	0.0	<.001
Gemini	956.5	<.001	704.5	<.001	162.5	<.001
GPT	27.0	<.001	24.5	<.001	381	<.001

Table 7: Wilcoxon signed-rank test results for each model against the human composite mean across three dimensions.

Configuration	Foreign Accentedness		Comprehensibility		Intelligibility	
	ICC2	ICC2k	ICC2	ICC2k	ICC2	ICC2k
human raters (baseline)	.209	.725	.336	.835	.493	.907
+Qwen2	.122	.605	.219	.755	.442	.897
+Qwen2-Instruct	.187	.716	.292	.819	.429	.892
+Ultravox	.174	.698	.235	.771	.168	.690
+Gemini	.178	.704	.289	.817	.391	.876
+GPT	.147	.655	.273	.805	.425	.891

Table 8: ICC2 (single rater) and ICC2k (average raters) when each model is added as an 11th rater to the human panel.

Rater	INT <i>r</i>	INT MAE	COMP <i>r</i>	COMP MAE	FA <i>r</i>	FA MAE
R01	.699	8.64	.660	0.76	.650	0.94
R02	.747	7.75	.587	1.41	.491	1.45
R03	.688	8.56	.577	1.07	.526	1.09
R04	.736	8.02	.473	1.37	.397	1.26
R05	.756	7.85	.645	1.06	.392	1.41
R06	.714	10.27	.438	1.56	.499	1.32
R07	.714	8.77	.602	1.39	.430	0.94
R08	.743	7.21	.534	1.45	.435	1.74
R09	.723	10.64	.597	1.04	.494	1.18
R10	.726	7.17	.564	1.34	.317	1.10
Mean	.725	8.49	.568	1.24	.463	1.24

Table 9: Individual human rater leave-one-out (LOO) correlations and mean absolute error across dimensions. INT, COM, and FA refer to intelligibility, comprehensibility and foreign accentedness, respectively.

	Mean diff	SD	LoA lower	LoA upper	Range	W
Intelligibility — Open	21.09	32.76	85.29	+43.11	128.40	508.0
Intelligibility — Closed	+11.61	13.84	15.52	+38.73	54.25	381.5
Comprehensibility — Open	+0.97	2.67	4.26	+6.20	10.47	1144.0
Comprehensibility — Closed	1.73	1.75	5.17	+1.71	6.87	704.5
Foreign-accentedness — Open	0.65	2.59	5.72	+4.42	10.14	19.5
Foreign-accentedness — Closed	1.10	2.34	5.68	+3.49	79.17	956.5

Table 10: Bland-Altman statistics

Incentives Of EdTech: A Systematic Review Of EduNLP Research

Gabrielle Gaudeau¹, Aoife O’Driscoll¹, Jasper Degraeuwe²,
Andrew Caines¹, Donya Rooein³, Zeerak Talat⁴

¹ALTA Institute, Computer Laboratory, University of Cambridge (UK),
²Ghent University (Belgium), ³Bocconi University (Italy), ⁴University of Edinburgh (UK)

Correspondence: gjg34@cam.ac.uk, ao514@cam.ac.uk, jasper.degraeuwe@ugent.be,
apc38@cam.ac.uk, donya.rooein@unibocconi.it, z@zeerak.org

Abstract

The global teacher shortage is pushing schools and institutions towards an ever-greater reliance on artificial intelligence. While the Natural Language Processing community has dedicated significant resources in developing educational technologies (EdTech) that support this shift, it remains unclear whose interests are being best served among the stakeholders of education.

In this paper, we present a systematic literature review of 204 papers published in venues of the Association for Computational Linguistics’ Special Interest Group on Building Educational Applications in 2024 and 2025, and validate these against EdTech papers from the wider ACL Anthology. By examining stakeholder inclusion and the prioritisation of research tasks, our findings reveal a critical tension: a push and pull between private-sector incentives and the foundational needs of educational infrastructure. Our analysis reveals that teachers are systematically under-represented as beneficiaries of research (33.3%) despite being the most affected, that real-world deployment remains rare (9.8%), and that ethical engagement tends toward acknowledgement rather than action. Drawing on exemplary papers in our corpus, we offer concrete recommendations for more responsible EduNLP research practices.

1 Introduction

Education has long been a domain of inspiration for Artificial Intelligence (AI) and Natural Language Processing (NLP). From early feature-based auto-markers (e.g., *e-rater*[®]; Attali and Burstein, 2006) to large language model (LLM)-powered intelligent tutoring systems (ITS) (e.g., Khanmigo¹ by Khan Academy), the goals have remained constant: for technology to extend the reach of good teaching and to support learners who might otherwise go without. These are meaningful goals –

socially urgent, technically challenging, and worthy of scientific investment – and their urgency has only grown in recent years with global teacher shortages (UNESCO, 2026), widening equity gaps (World Inequality Lab, 2026), and the rapid uptake of commercial AI products for education (Gomes, 2026). Held together, they have made the question of the role of technology in supporting education more pressing than ever.

There is a particular risk that comes with being deeply embedded in a fast-moving research area: the closer we are to the technical problems in front of us, the easier it is to lose sight of the overarching goal. As researchers, we are drawn towards the datasets we know, the metrics we trust, the tasks where progress is legible. Specialisation is necessary, but it can quietly narrow the frame of reference until the question, “Does this system work?”, crowds out the most important question: “Does this actually serve the people we said we were building it for?” This paper is, in part, an attempt to step back from that narrowing and ask plainly: as a field, are we meeting our own aspirations?

To answer this question, we conduct a systematic literature review of EduNLP research. We survey 204 papers published in 2024 and 2025 at ACL SIGEDU venues (BEA² and NLP4CALL³ workshops) and the main *ACL conferences. To the best of our knowledge, this is the first systematic review of EduNLP research that focuses on publications in the ACL Anthology. For each paper, we examine its tasks, motivations, stakeholder inclusion, incentives, and engagement with ethical risks to answer three research questions:

RQ1 Which tasks are prioritised in EduNLP research, what motivates them, and in which contexts are the resulting systems deployed?

¹<https://www.khanmigo.ai>

²Workshop on Innovative Use of NLP for Building Educational Applications

³Workshop on NLP for Computer-Assisted Language Learning

RQ2 Who are the stakeholders of EduNLP research, how are they included, and whose interests does the research serve?

RQ3 What risks, concerns, and limitations are raised, and to what extent does the research mitigate them?

Our findings show that teachers are systematically under-represented as beneficiaries in EduNLP research, real-world deployment is rare, and ethical engagement tends toward acknowledgement rather than action. We identify exemplary counter-examples and derive from them a set of concrete recommendations for the field.

2 Related Work

Education has been a domain for innovation dating back millennia. Digital technology is a modern feature of this long history: much of the early pioneering work on AI in the twentieth century was directed towards educational aims and applications in AIED (Newell et al., 1958; Minsky, 1974; Papert, 1980; Doroudi, 2023). In recent years the growth of interest in LLMs has also seen increasing application to education (Caines et al., 2023; Davis et al., 2024; Pack et al., 2024), further evidenced by the growing popularity of the annual Workshop on Innovative Use of NLP for Building Educational Applications (BEA), the foundation of the ACL SIGEDU in 2017,⁴ and investment by large technology firms into products such as Google’s LearnLM⁵ and OpenAI’s ChatGPT Edu.⁶

EdTech covers a wide-range of applications for educational purposes, often involving AI or NLP. There have been several surveys on EdTech and its use in various domains (Ahmad et al., 2024; Benedetto et al., 2023; Hidayat and Firmanti, 2024) spanning classroom support, virtual learning environments, websites, and tutoring chatbots. In this paper, we focus on ethical matters, which have received growing attention in AI and NLP more broadly, including the identification of different bias types throughout the “machine learning life cycle” (Suresh and Gutttag, 2021).

Within EdTech, several surveys and position papers have addressed ethical issues. Yan et al. (2025) presents a systematic review of 34 publications involving EdTech with AI in schools or higher education from 2020-2024, reporting a “constellation

of recurring ethical tensions” relating to algorithmic bias, data privacy, transparency, accountability, and academic integrity. They observe that these are known issues with AI applications, and recommend co-design with stakeholders, an emphasis on explainability, regulatory improvements, and AI literacy training for teachers. Alfredo et al. (2024) arrive at similar conclusions from a review of 108 papers relating to human-centred or participatory design and learning analytics.

Fu and Weng (2024) conduct a systematic review of empirical studies focused on EdTech and responsible AI, making similar conclusions to Yan et al. (2025) based on 40 selected papers. They present a vision for “responsible human-centered AIED” which includes core principles of Fairness and Equity, Transparency and Intelligibility, Agency and Autonomy, Privacy and Security, and Beneficence and Non-maleficence. Holmes et al. (2022) surveyed EdTech researchers, reporting high interest in but low confidence about ethical issues, attributed to a lack of ethics training in AI-related courses. They propose a framework for ethics in AIED aimed at ensuring “ethical by design” research, and emphasise the importance of cross-disciplinary engagement. Taken together, these reviews converge on a shared diagnosis: ethical considerations are widely recognised in principle but inconsistently integrated in practice.

This review extends these prior work by including research published throughout 2025, and by considering tasks, contexts, stakeholders, incentives, and risks across 204 EduNLP papers from *ACL main conference and workshop proceedings.

3 Methodology

Search Protocol. We collected all papers from the BEA and NLP4CALL workshops published in 2024 and 2025. We also conducted a search of the ACL Anthology using the Anthology API⁷ for papers published in main *ACL and associated conferences whose title or abstract contained at least one of 38 EduNLP-relevant search terms (e.g., “student modeling”; see Appendix B for a complete list of venues and search terms).⁸ This sampling approach affords an in-depth view into contemporary trends at the expense of longitudinal analyses. This search resulted in 191 papers from the two workshops, and 316 papers from *ACL conferences.

⁴<https://sig-edu.org/>

⁵<https://cloud.google.com/solutions/learnlm>

⁶<https://openai.com/chatgpt/education/>

⁷<https://acl-anthology.readthedocs.io/py-v0.5.3/api/>

⁸The search was conducted on January 21, 2026.

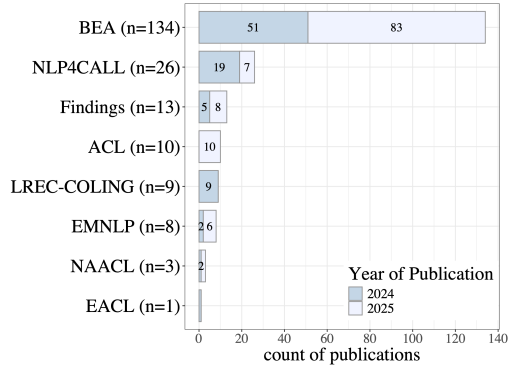


Figure 1: Number of papers per venue and year. We reviewed a total of 204 papers (160 BEA+NLP4CALL papers and 44 ACL Anthology main conference papers).

For BEA and NLP4CALL, we randomly sample 25% of contributions for each shared task, with a minimum sampling threshold of 5 papers for each task. We further include all shared task overview papers, as these represent a qualitatively distinct type of contribution. For the *ACL main conference papers, we reviewed all abstracts for relevance to educational applications, excluding 214 papers as non-relevant. The remaining 102 papers were stratified by publication year, venue, and search term, yielding a sample of 44 papers. This resulted in a final sample of 160 papers from BEA and NLP4CALL workshops, and 44 papers from *ACL conferences, for a total of 204 papers (see Table 8 for paper details). Figure 1 shows the distribution of papers across venues and years.

Data extraction. Data extraction was conducted manually by three of the authors using a shared extraction schema (see Appendix C). The schema captures: the specific task addressed; datasets used and their availability; the explicit motivation for the research; stakeholders mentioned and included (with associated quotes); the level of stakeholder inclusion; the deployment context of any system; incentives (both explicit and implicit) that the research serves; ethical risks and concerns raised; measures taken to address those risks; and future directions pertaining to risk, ethics, or aspiration.

Extraction proceeded in three phases. In the first phase (1), a single paper was annotated collaboratively to develop and validate the schema. In the second phase (2), annotators independently reviewed a shared batch of 25 papers,⁹ meeting to

⁹The shared batch was a stratified sample from our corpus of 204 papers (12.3%) based on venue and year of publication; it included 6 BEA 2024, 10 BEA 2025, 2 NLP4CALL 2024,

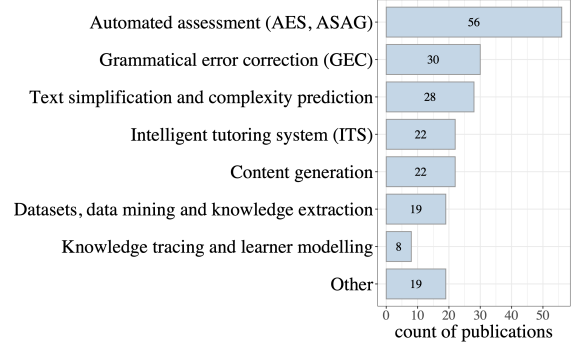


Figure 2: Number of papers per high-level task. See Table 8 for the detailed mapping.

discuss schema revisions and resolve ambiguities. Note that phase (2) was conducted in an iterative manner: following phase (1), each time the schema was modified or extended, all annotators updated their previous phase (2) annotation to reflect the revised guidelines. In the third and final phase (3), the remaining papers were reviewed independently by three authors. Extracting data took an annotator on average 45 minutes per paper (ranging between 30–60 minutes); we estimate that the review took a combined total of about 190 hours to complete.

Agreement. Inter-annotator agreement (IAA) was measured on phase (2)’s independently reviewed shared batch. Table 1 in Appendix D shows the agreement for the free-text dimensions of our schema based on the Percentage Agreement (PA; Roaché, 2017) measure (see Tables 6 and 7 for illustrations of how free-text agreement was computed). For the four multi-label dimensions, we report both Krippendorff’s α (Krippendorff, 2011) and PA in Tables 2, 3, 4 and 5.

For the free-text fields, PA ranges between 0.52 (for implicit incentives) and 1 (for deployment). For the multi-label dimensions, PA was consistently high (0.84–94 overall), while α was more variable. Agreement on the presence of stakeholders was generally moderate to strong ($\alpha = 0.49$ –0.7 overall, with agreement on teachers being particularly high at $\alpha = 0.79$ –0.84). Agreement on stakeholder inclusion level and risk engagement level was lower ($\alpha = 0.52$ –0.61 overall). Taking into account the qualitative and inherently interpretative nature of the annotation task (especially for dimensions such as risks/concerns), we consider these agreement values to be sufficiently high to

1 NLP4CALL 2025, 1 EACL 2024, 1 LREC-COLING, 1 NAACL 2025, 1 ACL 2025, and 2 Findings 2025 papers.

justify the independent reviewing in phase (3).

4 Tasks, Motivations, Deployment

Tasks. Figure 2 shows the distribution of high-level tasks across our corpus of papers. Automated assessment – i.e., automated essay scoring (AES) and automated short-answer scoring (ASAG) – is by far the most common task (56 papers), followed by grammatical error correction (GEC, 30 papers) and text simplification and complexity prediction (28). Content generation (22), intelligent tutoring systems (ITS, 22 papers), dataset creation and knowledge extraction (19), and knowledge tracing and learner modelling (8) are also represented. The “Other” task type includes a variety of research, most often relating to the novel capabilities of LLMs (e.g., multimodal assessment, alignment with human eye-tracking data, and discourse evaluation) and detecting LLM-generated texts.

The dominance of language assessment and feedback tasks is striking: taken together, AES/ASAG and GEC account for almost half of the corpus. This reflects a longstanding priority in EduNLP: indeed, automated assessment has been an active area of research for decades, benefitting from well-established datasets (e.g., ASAP; Hamner et al., 2012). However, this prevalence also raises questions about whose priorities are being served: automated assessment and feedback tools are of direct commercial value to large-scale testing organisations and EdTech companies.

Shared tasks. The NLP4CALL 2025 shared task introduced multilingual GEC (Masciolini et al., 2025), a direction of particular importance given that GEC, while already the second most represented task in our corpus, has historically been dominated by English-language systems. Broadening GEC to multilingual settings introduces non-trivial challenges around low-resource languages, cross-lingual transfer, and the availability of annotated learner corpora, and a shared task framing is well-suited to mobilising community effort around these barriers. On the other hand, the BEA 2024 shared tasks addressed automated prediction of item difficulty and response time (Yaneva et al., 2024a), and multilingual lexical simplification (Shardlow et al., 2024); the 2025 shared task addressed pedagogical ability assessment of AI-powered tutors (Kochmar et al., 2025). We note that all three of these problems receive less attention in the non-shared-task literature.

This suggests that shared tasks are playing a valuable role in broadening the community’s agenda, including towards less commercially obvious but educationally important problems such as pedagogical quality assessment, and towards underserved languages in otherwise established tasks. Beyond their immediate proceedings, shared tasks also exert a longer-lasting influence through the datasets they produce; resources like the W&I+LOCNESS dataset which was introduced for the BEA 2019 Shared Task on GEC (Bryant et al., 2019) tend to attract sustained reuse by the community (as illustrated by Figure 3), and thus continue to shape which problems remain visible and tractable long after the shared task itself has concluded.

Datasets. Papers in the corpus reported using 284 distinct datasets used a combined total of 460 times (373 for public datasets, 33 for those available upon-request and 54 for private datasets). Figure 11 shows that 73.9% of datasets used are publicly available, 7.4% are only available upon-request or through paid licences, and 18.7% are private. While the high proportion of public datasets is a positive indicator for reproducibility, Figure 3 reveals a high concentration of usage around a small number of datasets: the top three – W&I+LOCNESS (Bryant et al., 2019), ASAP (Hamner et al., 2012), and CoNLL-2014 (Ng et al., 2014) – together account for 12.9% of total public dataset usage (373), with a long tail of datasets used only once. This concentration partially reflects the task distribution noted previously: namely that AES and GEC are both well-established. However, this also raises questions about whether research findings generalise beyond the narrow slice of learner populations, languages, and educational contexts that these datasets represent. We return to this concern in Section 7.

Motivations. During extraction, we took note of the explicit motivation presented by papers for their presented research, and later classified each into one or more of seven high-level categories. Figure 12 shows that the most common motivation type across our corpus is to “help a stakeholder” (110 papers), followed by addressing a pedagogical or ethical concern (82), and assuming the role of a stakeholder (53). Technical motivation alone, with no stated stakeholder benefit, accounts for 43 papers, which is a non-trivial proportion (21.1%). Figure 4 reveals the stakeholder composition underlying papers’ motivations: learners and students are

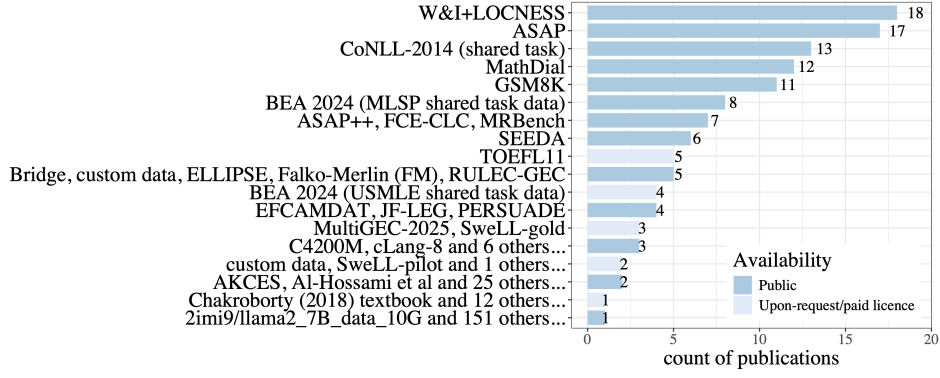


Figure 3: Dataset popularity (i.e., the number of times a dataset was used, and not only mentioned). We do not report private datasets given their absence of references.

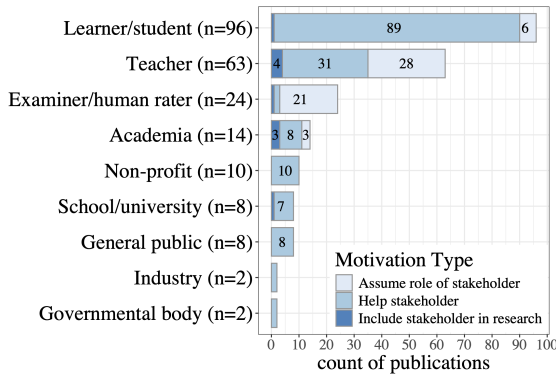


Figure 4: Distribution of different stakeholders for the three stakeholder-based motivations in Figure 12.

invoked in 61.1% of papers with stakeholder-based motivation (96 of 157 papers), making them by far the most frequently cited intended beneficiary. Teachers appear in 40.1% of such papers (63 of 157 papers), though they are most commonly invoked as a pressure points, referenced in terms of the cost, time, or burden associated with their labour, and implicitly positioned as a bottleneck that automation should relieve. This framing matters. A motivation to reduce teacher burden through automation is meaningfully different from one that seeks to augment teacher capability or support teacher agency. In a number of papers in our corpus, teachers appear in the motivation but then disappear from the research design entirely: they are not consulted, included in evaluation, or named as beneficiaries of the results. We discuss this pattern and its implications further in Section 6.

Context deployment. Figure 5 shows that 79.4% of papers (162 papers) present systems or models that are never deployed to real-world users. Only

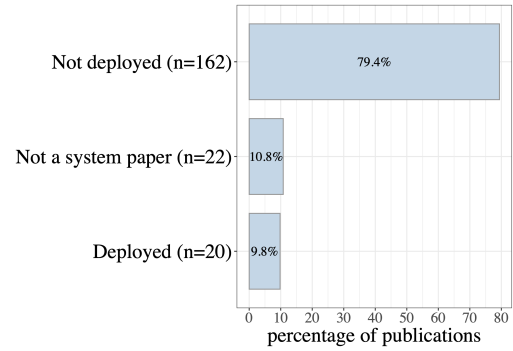


Figure 5: Papers that deployed their method to real-world users or tested it on pre-existing real-world data.

9.8% of papers report genuine deployment. We label resource and survey papers as “Not a system paper.” Non-deployment is not itself a failing: fundamental research that develops methods, datasets, or evaluation frameworks may legitimately precede any deployment. More concerning is that papers describing non-deployed systems rarely discuss the pathway to deployment: the educational contexts in which the system might operate, the stakeholders who would need to be involved, or the risks real-world deployment would introduce. This creates a body of research that is optimised for benchmark performance in conditions that may bear little resemblance to the classrooms, tutoring sessions, and assessment environments it nominally serves.

5 The Roles of Stakeholders

Author affiliations and acknowledged entities.

Figure 15 shows that the paper author affiliations in our corpus are geographically concentrated: the United States accounts for the largest single-country share of author affiliations (58 papers), followed by Germany (29 papers), China (23 papers),

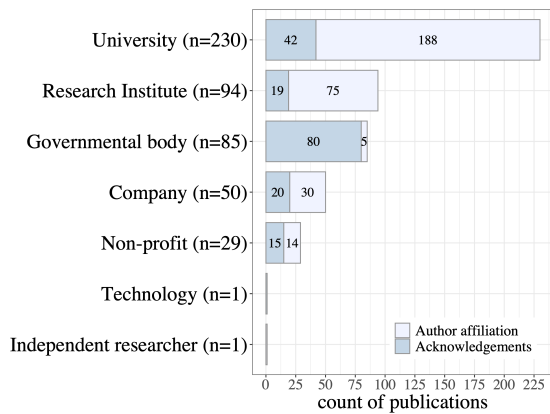


Figure 6: Number of papers per type of author affiliation and type of entity mentioned in acknowledgements.

and other European countries (similar observations can be made on the origin of the acknowledged entities in Figure 14). Figure 6 shows that universities dominate author affiliations (188 papers), followed by research institutes (75) and companies (30). Funding acknowledgements are concentrated within governmental bodies (80), with national science foundations of China and the US appearing the most frequently (Figure 16). Industry acknowledgements (e.g., Microsoft) appear in a small but non-trivial number of papers (20). While industry involvement in research funding is not inherently problematic, it creates potential conflicts of interest that deserve explicit discussion, particularly in a field where commercial EdTech products are directly shaped by research agendas. Notably, few papers in our corpus explicitly disclose or discuss potential conflicts of interest arising from their funding sources; a gap that mirrors findings in adjacent fields (Garrett et al., 2020).

Stakeholders mentioned or included. Figure 7 shows that learners and students are mentioned in the most papers overall (170 papers), followed by teachers (97), and domain experts (88). However, mention does not equate to inclusion: the proportion of mentioned stakeholders who are also actively included in the research is substantially lower across all groups. Among teachers, 26.8% of papers that mention them also include them in the research (26). For learners, 22.4% of mentioning papers include them (38). Domain experts show a much higher inclusion rate (56.8%), in part because they are frequently recruited as annotators or raters. Most strikingly, parents were only mentioned in two papers, despite their having such an important

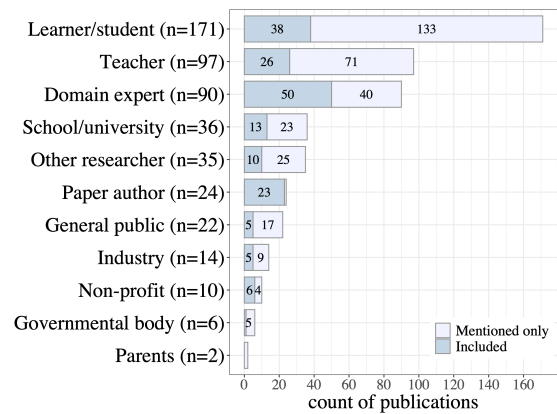


Figure 7: Number of papers per type of stakeholder included or mentioned only in the research.

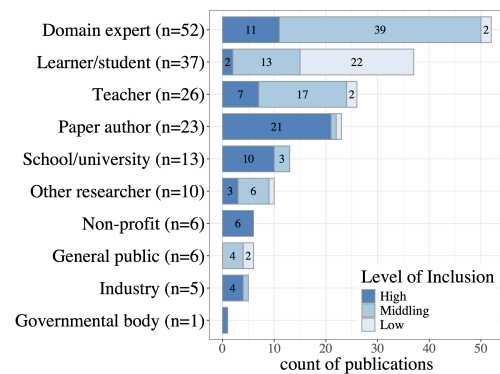


Figure 8: Level of inclusion of included stakeholders by stakeholder type; we distinguish 3 levels: *High* (integral to research design & completion), *Middling* (involved in data evaluation or annotation, without input on research design), and *Low* (test subjects in data collection only).

role in children education (Kostov, 2026).

Figure 13 reveals the overall distribution of inclusion levels across all included stakeholders: 47.0% of inclusions are classified as *Middling* (involved in data evaluation or annotation, but with no input on research design), 32.1% as *High* (integral to research design and completion), and 20.9% as *Low* (test subjects in data collection only). Figure 8 shows that this breakdown varies substantially by stakeholder type. Other than paper authors themselves, schools and universities are most likely to be included at a *High* level (76.9%), while teachers, when included at all, are predominantly included at a *Middling* level (65.5%), most often as annotators. Learners are most often included as test subjects (59.5%). The implication is that even when stakeholders are formally included, they are rarely positioned as agents who shape the research, they are more often positioned as instruments of it.

Incentives. Figure 9 shows the distribution of stakeholders explicitly mentioned as benefiting from the research alongside those we identified as implicit beneficiaries. We note that the identification of implicit beneficiaries is the most subjective dimension of our annotation: it required annotators to infer who stands to gain from a piece of research beyond what authors themselves state, based on the nature of the task, the deployment context, and the funding sources involved. For instance, a paper developing an AES system for standardised testing, funded by a testing organisation, was coded as implicitly benefiting industry, even if no such benefit was named. Due to the subjective nature of this dimension, inter-annotator agreement was accordingly lower (0.53; Table 1), and these findings should be read as indicative rather than definitive.

Learners and students are the most frequently named explicit beneficiary (125 papers). Teachers stand out starkly here: 80.9% of their appearances are explicit (55 papers). Stated differently, teachers are almost never the unstated but evident beneficiary of research; when they benefit, papers say so. However, the vast majority of papers do not position them as benefiting at all. On the other hand, non-profit organisations, industry and governmental bodies appear prominently as implicit beneficiaries. That is, while they are not named in the paper as intended beneficiaries, the research clearly serves their interests. This is most visible in the task-level breakdown in Figure 17: automated assessment research (the largest task category in the corpus) consistently benefits learners and industry, while teachers and examiners are sparsely represented. The commercial relationship here is direct: automated scoring tools reduce the need for human markers and are of clear value to large-scale testing organisations. For ITS, learners dominate, with limited acknowledgement of teachers. GEC research shows the broadest stakeholder spread, in part because GEC tools serve not only learners and teachers but also the general public who use writing assistance tools in everyday tasks.

6 Risks, Concerns, Limitations, and Measures Taken

Risks, concerns and limitations raised. Figure 18 shows the distribution of risks, concerns, and limitations explicitly raised by paper authors, organised into six high-level categories. We note that inter-annotator agreement was lower for this

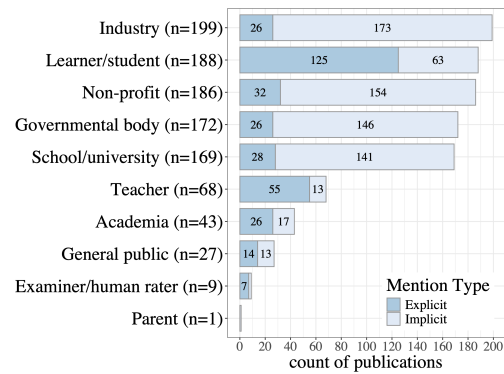


Figure 9: Stakeholders explicitly stated as benefitting from the research, as well as those that we could see benefitting that were not explicitly mentioned (*Implicit*). Note that a stakeholder may be both explicitly mentioned to benefit in some way and implicitly in another.

dimension than others (0.57; Table 1), owing to the need to assess coverage across a large and varied set of concerns; these results should therefore be read as indicative trends rather than precise counts. The most commonly noted concerns are methodology limitations (69 papers), dataset limitations (60), followed by lack of generalisability and language-specificity (56), risk of bias (46) and and task/domain-specific limitations (44), reflecting the tendency of research to develop systems for specific languages or educational contexts that may not transfer. Several important risk categories are raised much less frequently. Risk of hallucination appears in only 12 papers, risk of dual-use in 6, and safety concerns in 26. Within the contextualising research category, the gap between research and real-world application is noted in 32 papers and the need for human-in-the-loop in 19, suggesting some awareness of deployment limitations, this rarely translates into direct mitigation (Figure 19). Data protection and anonymisation concerns are raised in 37 papers, while informed consent and fair compensation for included stakeholders, critical ethical requirements for human-subjects research, appear in only 11 and 10 papers respectively. That human-subjects protections remain among the least commonly raised concerns in a corpus that routinely collects learner data and recruits human annotators is itself a notable finding.

Engagement with risks. Figure 19 distinguishes three levels of engagement with stated risks: *High* (directly mitigated or discussed in substantial depth), *Middling* (discussed as part of future work), and *Low* (briefly mentioned only). Across most risk

categories, the majority of engagement is at a *Low* or *Middling* level. *High* engagement is most consistently found in the participant and data concern category: fair compensation for included stakeholders (100.0%) and informed consent (72.7%) are the most actively addressed concerns, though both are raised by relatively few papers to begin with. By contrast, the largest categories show the weakest engagement: methodology limitations are 98.6% *Middling* or *Low*, and dataset limitations 90.0% *Middling* or *Low*. Risk of bias, one of the most frequently raised concerns at 46 papers, is engaged at a *High* level in only 15.2% of cases. The gap between research and real-world application and the need for human-in-the-loop, two concerns with clear implications for responsible deployment, are predominantly *Middling* or *Low*. This pattern suggests a community that is aware of the ethical dimensions of its work but has not yet developed consistent norms for acting on them within the scope of individual papers.

Future work. Figure 20 shows a distribution of the areas of future work explicitly mentioned in the papers. We report future work specifically related to any risks, concerns or higher aspirations rather than any purely technical work; of our data sample, 21 papers do not discuss any such future work. Four high-level themes emerge within discussed future work: stakeholder inclusion, technical development, expanding the scope of the research, and engaging with issues emerging from the research. Of these, the most frequently mentioned category is expanding the scope of the research, with expanding the data (42), language selection (36), and subject domain (35) the most common fine-grained directions. EduNLP research is often performed at language- or task-specific levels, resulting in common limitations which translate to clear future directions. The least common high-level category is engaging with issues emerging from research, with fine-grained categories including interpretability and bias mitigation (16 and 14 papers respectively), exploring performance-cost trade-offs (12), and initiating broader discussions in the EduNLP space. Within the fine-grained categories overall, the most frequently referenced future direction is general technical improvements related to the paper’s risks and concerns (91 papers). Despite controlling for purely technical work in our analysis, the primary focus for EduNLP researchers remains within this domain. Within the stakeholder inclusion category,

user study and user inclusion (37) and integration into real-world systems (32) are the most common directions, suggesting some awareness that current work falls short. Analysis of the future directions in our sample therefore reveals a tendency towards prioritising empirically-motivated fine-grained technical work rather than ethically-driven broader work. In part, this may be due to an imbalance in available resources for conducting such research.

7 Discussion: Opportunities, Recommendations, Aspirations

Opportunities. Our findings reveal some structural gaps in EduNLP research that constitute genuine opportunities for the field. First, teachers are under-represented both as beneficiaries and active participants, despite their central role in education. This represents a significant misalignment between stated purpose and actual design. Research that nominally aims to support education but systematically excludes the professional educators who mediate it risks building tools that are technically sophisticated but pedagogically ill-fitting, or that automate away precisely the human judgement that makes good teaching effective. Second, the gap between research development and real-world deployment is striking: only 9.8% of system papers are deployed in live educational settings. This reflects a missing discourse about what responsible deployment looks like: which stakeholders need to be involved, what evaluation is appropriate for real students and teachers, and what accountability mechanisms should be in place. This gap is further sharpened by the concentration of datasets around high-stakes standardised testing contexts, and by the dominance of assessment tasks which, given their direct commercial value to the testing industry, risk pulling the research agenda toward institutional efficiency over the full range of educational stakeholders.

Recommendations. Drawing on exemplary papers in our corpus, we offer three concrete recommendations for the EduNLP community:

1. **Co-design with teachers and learners from the outset.** Research that positions stakeholders as genuine co-designers, rather than test subjects or future-work items, produces better-grounded systems and more honest evaluation. Galletti and Cesaroni (2025) offer a replicable model for this: conducting focus

groups and questionnaires with teachers at different stages of system development surfaced concerns around transparency, autonomy, and pedagogical alignment that would not have emerged from technical evaluation alone. See also [Huovinen and Hämäläinen \(2025\)](#). Their work echoes principles of design justice ([Costanza-Chock, 2020](#)) which seek to decentre technical expertise in favour of lived experience and domain expertise – in all regards save technical implementation – as a mechanism for ensuring that those affected by a system retain meaningful agency in shaping it. As it stands, a true expression of design justice was not found in any of the reviewed papers of this corpus, however [Wang et al. \(2025c\)](#) embodies some aspects of it. Though not a system, the paper demonstrates that design justice principles can be embedded even at the resource creation stage: their math world problem benchmark was developed through structured interviews with primary school math teachers, whose pedagogical expertise directly shaped what counts as a meaningful visual, ensuring that future systems trained or evaluated on this benchmark will be held to a standard defined by them.

2. **Make deployment contexts and costs explicit.** Authors should describe the educational context in which their system could or has been deployed, the stakeholder roles involved, and provide an honest account of computational, financial, and human costs alongside claimed benefits ([Akter et al., 2025](#); [Gupta et al., 2025](#); [Li and Ng, 2024](#)).
3. **Adopt structured ethical reflection and act on it.** Our data show that named concerns rarely translate into mitigation within the same paper. Venues should normalise the expectation that ethical risks raised are addressed in the current work, not deferred to future work. Checklists like the ARR Responsible NLP Checklist already support this: they prompt authors to interrogate their own design choices (e.g., [Goto et al., 2025a](#), who voluntarily engage in detailing a number of the checklist items) and give reviewers a structured basis for evaluating ethical engagement.

Aspirations. The potential for AI in education is genuine: it could improve access to education,

helping reduce inequalities related to geography, language, resources, and infrastructure, for learners who might otherwise go without. It could also help free educators of repetitive and time-consuming tasks so they can concentrate on the relational aspect of education that systems cannot and should not replace. Automated tools can also mitigate some human weaknesses that threaten fairness in assessment: fatigue, inconsistency, and unconscious biases. The question is not whether AI belongs in education, but whether we are stewarding its development responsibly. The exemplary papers in our corpus demonstrate that it is possible. Our aspiration for the field is a research community that treats educational infrastructure as a site of social responsibility, not merely technical opportunity. The trajectory the field takes will depend on choices that are made now about which tasks to prioritise, whose voices to include, and what counts as success. [Harding \(2025\)](#), reflecting on AI in language assessment, frames this as a choice between utopian and dystopian futures: one in which assessment technology is context-sensitive, transparent, connected with learning, and deeply oriented toward justice; as opposed to one driven by expediency, opacity, and the logic of scale.

8 Conclusion

This paper has presented a systematic review of 204 EduNLP papers published at ACL SIGEDU venues and main *ACL conferences in 2024 and 2025, examining tasks, motivations, stakeholder inclusion, incentive structures, and ethical engagement. Our analysis reveals a field that is technically productive but structurally misaligned with key educational stakeholders, particularly teachers who are rarely included in research and almost never positioned as implicit beneficiaries. At the same time, our corpus contains exemplary work that demonstrates what responsible, stakeholder-grounded EduNLP research looks like in practice. The norms and practices embedded in these papers are neither technically burdensome nor novel in principle. What is needed is for the community to adopt them consistently, and for publication venues to create the conditions in which doing so is expected rather than exceptional. We hope this review serves as both a diagnostic and a resource: a map of where the field currently stands, and a set of orientations for where it should go.

9 Limitations

This review has several limitations that should be noted. First, while our corpus of 204 papers is broad in scope, it is not exhaustive, meaning that some relevant papers will have been missed. Our focus on ACL Anthology venues also means that work published in AIED journals, learning analytics conferences, and EdTech-specific venues falls outside our scope: the picture we paint is of the NLP community specifically, not the broader field. Second, annotation of inherently interpretive dimensions, particularly stakeholder inclusion level and risk engagement level, carries subjectivity that agreement scores can only partially represent. We report these as indicative trends rather than precise counts, but readers should bear this in mind when interpreting figures. Third, our corpus covers 2024–2025 only; while this captures the most recent work, it is a short window and trends may not generalise to earlier or future periods. An interesting direction for future work would be to extend the temporal scope to publications published before the release of ChatGPT (OpenAI, 2026) in November 2022, which would allow for a direct comparison of research priorities, stakeholder inclusion, and ethical engagement before and after the widespread availability of generative AI. Finally, as researchers embedded in the EduNLP community ourselves, we are not neutral observers, our framing of what constitutes meaningful stakeholder inclusion or adequate ethical engagement reflects our own values, which we have tried to make explicit throughout.

On financial disclosures. While complete financial disclosures of which entities have funded research is a desirable trait in papers due to the transparency it affords, disclosure can be structurally limited. For example, some grant funders – particularly military funding – may require non-disclosure, nation-wide regulation may limit disclosure, research can be funded across multiple grants, work may be conducted on an entirely voluntary basis, among many other reasons. We therefore see financial disclosure as a spectrum between complete opacity and complete transparency. We advocate for researchers to approach the question of financial disclosure according to a maximalist approach, i.e., we argue that researchers should share as much information as is possible to them in a given situation. A transparency maximalist approach will afford greater insight into how research into educational technologies is being, forcefully and subtly,

shifted by the interests of different entities.

10 Ethical considerations

All papers surveyed in this review are publicly available through the ACL Anthology; no private or unpublished materials were used. No human subjects were involved in the review itself. The annotation process involved researchers reading and characterising the work of others, which carries a risk of misrepresentation; we have sought to mitigate this through iterative schema development, inter-annotator agreement measurement, and the use of direct quotes to ground our characterisations. Our normative claims – that teachers are under-served, that ethical engagement is insufficient, that commercial incentives distort research agendas – are recommendations and observations, not accusations about individual papers or authors. We acknowledge that we are ourselves part of the community we critique, and that future reviews may find similar gaps in our own work. This paper has been pre-registered on OSF.¹⁰

Acknowledgements

Gabrielle Gaudeau, Aoife O’Driscoll and Andrew Caines are supported by Cambridge University Press & Assessment. Donya Rooein is a member of the MilaNLP group and the Data & Marketing Insights Unit of the Bocconi Institute for Data Science and Analysis. Her research is supported through the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (No. 949944, INTEGRATOR). We thank the anonymous reviewers for their time and valuable feedback. Finally, we note that Claude Sonnet 4.6¹¹ was used to improve the language and readability of the manuscript.

References

- Kashif Ahmad, Waleed Iqbal, Ammar El-Hassan, Junaid Qadir, Driss Benhaddou, Moussa Ayyash, and Ala Al-Fuqaha. 2024. [Data-driven artificial intelligence in education: A comprehensive review](#). *IEEE Transactions on Learning Technologies*, 17:12–31.
- Soroosh Akef, Detmar Meurers, Amália Mendes, and Patrick Rebuschat. 2025. [Interpretable machine learning for societal language identification: Modeling English and German influences on Portuguese](#)

¹⁰https://osf.io/nhb2q/overview?view_only=533ca23658d644a4abaf0bbd7e63087c

¹¹<https://www.anthropic.com/claude/sonnet>

- heritage language. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 50–62, Tallinn, Estonia. University of Tartu Library.
- Syeda Sabrina Akter, Seth Hunter, David Woo, and Antonios Anastasopoulos. 2025. [Costs and benefits of AI-enabled topic modeling in P-20 research: The case of school improvement plans](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 460–476, Vienna, Austria. Association for Computational Linguistics.
- Riordan Alfredo, Vanessa Echeverria, Yueqiao Jin, Lixiang Yan, Zachari Swiecki, Dragan Gašević, and Roberto Martinez-Maldonado. 2024. [Human-centred learning analytics and AI in education: A systematic literature review](#). *Computers and Education: Artificial Intelligence*, 6:100215.
- David Alfter. 2024. [Out-of-the-box graded vocabulary lists with generative language models: Fact or fiction?](#) In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 1–19, Rennes, France. LiU Electronic Press.
- David Alfter. 2025. [The need for truly graded lexical complexity prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 326–333, Vienna, Austria. Association for Computational Linguistics.
- Bashar Alhafni and Nizar Habash. 2025. [Enhancing text editing for grammatical error correction: Arabic as a case study](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17892–17914, Vienna, Austria. Association for Computational Linguistics.
- Mina Almasi and Ross Deans Kristensen-McLachlan. 2025. [Alignment drift in CEFR-prompted LLMs for interactive Spanish tutoring](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 70–88, Vienna, Austria. Association for Computational Linguistics.
- Jiyuan An, Xiang Fu, Bo Liu, Xuquan Zong, Cunliang Kong, Shuliang Liu, Shuo Wang, Zhenghao Liu, Liner Yang, Hanghang Fan, and Erhong Yang. 2025. [BLCU-ICALL at BEA 2025 shared task: Multi-strategy evaluation of AI tutors](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1084–1097, Vienna, Austria. Association for Computational Linguistics.
- Aitor Arronte Alvarez and Naiyi Xie Fincham. 2025. [Automated L2 proficiency scoring: Weak supervision, large language models, and statistical guarantees](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 384–397, Vienna, Austria. Association for Computational Linguistics.
- Yuya Asano, Beata Beigman Klebanov, and Jamie Mikeska. 2025. [Exploring task formulation strategies to evaluate the coherence of classroom discussions with GPT-4o](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 716–736, Vienna, Austria. Association for Computational Linguistics.
- Nischal Ashok Kumar and Andrew Lan. 2024. [Improving socratic question generation using data augmentation and preference optimization](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 108–118, Mexico City, Mexico. Association for Computational Linguistics.
- Yigal Attali and Jill Burstein. 2006. [Automated essay scoring with e-rater® v.2](#). *The Journal of Technology, Learning and Assessment*, 4(3).
- Sarra El Ayari and Zhongjie Li. 2024. [Potential of ASR for the study of L2 learner corpora](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 49–58, Rennes, France. LiU Electronic Press.
- Nicolas Ballier and Adrien Méli. 2024. [Investigating acoustic correlates of whisper scoring for L2 speech using forced alignment with the Italian component of the ISLE corpus](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 20–32, Rennes, France. LiU Electronic Press.
- Stefano Bannò, Kate M. Knill, and Mark J. F. Gales. 2025. [Exploiting the English vocabulary profile for L2 word-level vocabulary assessment with LLMs](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 632–646, Vienna, Austria. Association for Computational Linguistics.
- Stefano Bannò, Hari K. Vydana, Kate M. Knill, and Mark J. F. Gales. 2024. [Can GPT-4 do L2 analytic assessment?](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 149–164, Mexico City, Mexico. Association for Computational Linguistics.
- Beata Beigman Klebanov, Michael Suhan, Tenaha O’Reilly, and Zuowei Wang. 2024. [From miscue to evidence of difficulty: Analysis of automatically detected miscues in oral reading for feedback potential](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 459–469, Mexico City, Mexico. Association for Computational Linguistics.
- 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.

- Luca Benedetto, Shiva Taslimipoor, and Paula Buttery. 2025. [A survey on automated distractor evaluation in multiple-choice tasks](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 55–69, Vienna, Austria. Association for Computational Linguistics.
- Santiago Berruti, Arturo Collazo, Diego Sellanes, Aiala Rosá, and Luis Chiruzzo. 2024. [Automatic cross-word clues extraction for language learning](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 381–390, Mexico City, Mexico. Association for Computational Linguistics.
- Marie Bexte, Yuning Ding, and Andrea Horbach. 2025. [Increasing the generalizability of similarity-based essay scoring through cross-prompt training](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 225–236, Vienna, Austria. Association for Computational Linguistics.
- Marie Bexte, Andrea Horbach, Lena Schützler, Oliver Christ, and Torsten Zesch. 2024. [Scoring with confidence? – exploring high-confidence scoring for saving manual grading effort](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 119–124, Mexico City, Mexico. Association for Computational Linguistics.
- Marie Bexte and Torsten Zesch. 2025. [Is lunch free yet? overcoming the cold-start problem in supervised content scoring using zero-shot LLM-generated training data](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 144–159, Vienna, Austria. Association for Computational Linguistics.
- Pramit Bhattacharyya and Arnab Bhattacharya. 2025. [Leveraging LLMs for Bangla grammar error correction: Error categorization, synthetic data, and model evaluation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8220–8239, Vienna, Austria. Association for Computational Linguistics.
- Louise Bloch, Johannes Rückert, and Christoph Friedrich. 2025. [Towards automatic formal feedback on scientific documents](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 334–344, Vienna, Austria. Association for Computational Linguistics.
- Stephen Bodnar. 2025. [A prototype authoring tool for editing authentic texts using LLMs to increase support for contextualised L2 grammar practice](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 63–71, Tallinn, Estonia. University of Tartu Library.
- Eujene Nikka V. Boquio and Prospero C. Naval, Jr. 2024. [Beyond canonical fine-tuning: Leveraging hybrid multi-layer pooled representations of BERT for automated essay scoring](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2285–2295, Torino, Italia. ELRA and ICCL.
- Allison Bradford, Kenneth Steimel, Brian Riordan, and Marcia Linn. 2024. [Building robust content scoring models for student explanations of social justice science issues](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 450–458, Mexico City, Mexico. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- 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)*, pages 522–527, Mexico City, Mexico. Association for Computational Linguistics.
- Andrew Caines, Luca Benedetto, Shiva Taslimipoor, Christopher Davis, Yuan Gao, Øistein Andersen, Zheng Yuan, Mark Elliott, Russell Moore, Christopher Bryant, Marek Rei, Helen Yannakoudakis, Andrew Mullooly, Diane Nicholls, and Paula Buttery. 2023. [On the application of large language models for language teaching and assessment technology](#). In *Proceedings of the Empowering Education with LLMs – the Next-Gen Interface and Content Generation Workshop at AIED*.
- Yayu Cao, Tianxiang Wang, Lvxiaowei Xu, Zhenyao Wang, and Ming Cai. 2025. [CxGGEC: Construction-guided grammatical error correction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6156, Vienna, Austria. Association for Computational Linguistics.
- Dan Carpenter, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng, and James Lester. 2024. [Assessing student explanations with large language models using fine-tuning and few-shot learning](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 403–413, Mexico City, Mexico. Association for Computational Linguistics.
- Abhirup Chakravarty, Mark Brenchley, Trevor Breakpear, Ian Lewin, and Yan Huang. 2025. [Enhancing marker scoring accuracy through ordinal confidence modelling in educational assessments](#). In *Proceedings of the 63rd Annual Meeting of the Association*

- for Computational Linguistics (Volume 6: Industry Track), pages 1498–1507, Vienna, Austria. Association for Computational Linguistics.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- Po-Kai Chen, Bo-Wei Tsai, Shao Kuan Wei, Chien-Yao Wang, Jia-Ching Wang, and Yi-Ting Huang. 2025. [Mixture of ordered scoring experts for cross-prompt essay trait scoring](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18071–18084, Vienna, Austria. Association for Computational Linguistics.
- Ruishi Chen and Yiling Zhao. 2025. [EduCSW: Building a Mandarin-English code-switched generation pipeline for computer science learning](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 908–919, Vienna, Austria. Association for Computational Linguistics.
- Yuan Chen and Xia Li. 2024. [PLAES: Prompt-generalized and level-aware learning framework for cross-prompt automated essay scoring](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12775–12786, Torino, Italia. ELRA and ICCL.
- Mihail Chifligarov, Jammila Laâguidi, Max Schellenberg, Alexander Dill, Anna Timukova, Anastasia Drackert, and Ronja Laarmann-Quante. 2025. [Automated scoring of a German written elicited imitation test](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 237–247, Vienna, Austria. Association for Computational Linguistics.
- Aymeric de Chillaz, Anna Sotnikova, Patrick Jermann, and Antoine Bosselut. 2025. [Challenges for AI in multimodal STEM assessments: a human-AI comparison](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 279–293, Vienna, Austria. Association for Computational Linguistics.
- Madalina Chitez, Liviu Dinu, Marius Micluta-Campeanu, Ana-Maria Bucur, and Roxana Rogobete. 2025. [Assessing critical thinking components in Romanian secondary school textbooks: A data mining approach to the ROTEX corpus](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 780–793, Vienna, Austria. Association for Computational Linguistics.
- SeongYeub Chu, Jong Woo Kim, Bryan Wong, and Mun Yong Yi. 2025a. [Rationale behind essay scores: Enhancing S-LLM’s multi-trait essay scoring with rationale generated by LLMs](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5811–5829, Albuquerque, New Mexico. Association for Computational Linguistics.
- Zhendong Chu, Jian Xie, Shen Wang, Zichao Wang, and Qingsong Wen. 2025b. [UniEDU: Toward unified and efficient large multimodal models for educational tasks](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1007–1016, Suzhou (China). Association for Computational Linguistics.
- Sofía Correa Busquets, Valentina Córdova Véliz, and Jorge Baier. 2025. [IALab UC at BEA 2025 shared task: LLM-powered expert pedagogical feature extraction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1187–1193, Vienna, Austria. Association for Computational Linguistics.
- Sasha Costanza-Chock. 2020. *Design justice*. The MIT Press.
- Petru 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)*, pages 610–617, Mexico City, Mexico. Association for Computational Linguistics.
- Scott Crossley, Perpetual Baffour, Mihai Dascalu, and Stefan Ruseti. 2024. [A world CLASSE student summary corpus](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 99–107, Mexico City, Mexico. Association for Computational Linguistics.
- Stefan Dascalescu, Marius Dumitran, and Mihai Alexandru Vasiluta. 2025. [Leveraging generative AI for enhancing automated assessment in programming education contests](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 89–99, Vienna, Austria. Association for Computational Linguistics.
- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipour, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.
- Kordula De Kuthy, Leander Gierbach, and Detmar Meurers. 2025. [Automatic concept extraction for learning domain modeling: A weakly supervised approach using contextualized word embeddings](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 175–185, Vienna, Austria. Association for Computational Linguistics.

- Michiel De Vrindt, Renske Bouwer, Wim Van Den Noortgate, Marije Lesterhuis, and Anaïs Tack. 2025. [Explaining holistic essay scores in comparative judgment assessments by predicting scores on rubrics](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 535–548, Vienna, Austria. Association for Computational Linguistics.
- Michiel De Vrindt, Anaïs Tack, Renske Bouwer, Wim Van Den Noortgate, and Marije Lesterhuis. 2024. [Predicting initial essay quality scores to increase the efficiency of comparative judgment assessments](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 125–136, Mexico City, Mexico. Association for Computational Linguistics.
- Jasper Degraeuwe. 2025. [You shall know a word’s difficulty by the family it keeps: Word family features in personalised word difficulty classifiers for L2 Spanish](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 312–325, Vienna, Austria. Association for Computational Linguistics.
- Jasper Degraeuwe and Patrick Goethals. 2024. [Leading by example: The use of generative artificial intelligence to create pedagogically suitable example sentences](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 33–48, Rennes, France. LiU Electronic Press.
- Yuning Ding, Julian Lohmann, Nils-Jonathan Schaller, Thorben Jansen, and Andrea Horbach. 2024. [Transfer learning of argument mining in student essays](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 439–449, Mexico City, Mexico. Association for Computational Linguistics.
- Kosuke Doi, Katsuhito Sudoh, and Satoshi Nakamura. 2024. [Automated essay scoring using grammatical variety and errors with multi-task learning and item response theory](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 316–329, Mexico City, Mexico. Association for Computational Linguistics.
- Shayan Doroudi. 2023. [The intertwined histories of artificial intelligence and education](#). *International Journal of Artificial Intelligence in Education*, 33:885–928.
- Marius Dumitran, Mihnea Buca, and Theodor Moroianu. 2025. [MateInfoUB: A real-world benchmark for testing LLMs in competitive, multilingual, and multi-modal educational tasks](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 24–37, Vienna, Austria. Association for Computational Linguistics.
- Matthew Durward and Christopher Thomson. 2024. [Evaluating vocabulary usage in LLMs](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 266–282, Mexico City, Mexico. Association for Computational Linguistics.
- Benjamin Dutilleul, Mathis Debaillon, and Sandeep Mathias. 2024. [ISEP_Presidency_University at MLSP 2024 shared task: 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)*, pages 605–609, Mexico City, Mexico. Association for Computational Linguistics.
- Mohamed Elaraby and Diane Litman. 2025. [Lessons learned in assessing student reflections with LLMs](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 672–686, Vienna, Austria. Association for Computational Linguistics.
- Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. [TRATES: Trait-specific rubric-assisted cross-prompt essay scoring](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. 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)*, pages 590–598, Mexico City, Mexico. Association for Computational Linguistics.
- Annie Foret, Erwan Hupel, and Pêr Morvan. 2024. [Enhancing a multi-faceted verb-centered resource to help a language learner: the case of breton](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 59–66, Rennes, France. LiU Electronic Press.
- Tania Amanda Nkoyo Frederick Eneye, Chukwuebuka Fortunate Ijezue, Ahmad Imam Amjad, Maaz Amjad, Sabur Butt, and Gerardo Castañeda-Garza. 2025. [Advances in auto-grading with large language models: A cross-disciplinary survey](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 477–498, Vienna, Austria. Association for Computational Linguistics.
- Yao Fu and Zhenjie Weng. 2024. [Navigating the ethical terrain of AI in education: A systematic review on framing responsible human-centered AI practices](#). *Computers and Education: Artificial Intelligence*, 7:100306.
- Martina Galletti and Valeria Cesaroni. 2025. [From end-users to co-designers: Lessons from teachers](#). In

- Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 505–516, Vienna, Austria. Association for Computational Linguistics.
- Natalie Garrett, Nathan Beard, and Casey Fiesler. 2020. **More than "if time allows": The role of ethics in AI Education.** In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, page 272–278. Association for Computing Machinery.
- Tianyi Geng and David Alfter. 2025. **Towards a real-time Swedish speech analyzer for language learning games: A hybrid AI approach to language assessment.** In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 186–201, Vienna, Austria. Association for Computational Linguistics.
- Dominik Glandorf and Detmar Meurers. 2024. **Towards fine-grained pedagogical control over English grammar complexity in educational text generation.** In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 299–308, Mexico City, Mexico. Association for Computational Linguistics.
- Ben Gomes. 2026. **Learners and educators are AI's new "super users".** *Google Blog: The Keyword*. [Accessed 31-03-2026].
- Dhiman Goswami, Kai North, and Marcos Zampieri. 2024. **GMU at MLSP 2024: Multilingual lexical simplification with transformer models.** In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 627–634, Mexico City, Mexico. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. **gec-metrics: A unified library for grammatical error correction evaluation.** In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. **Reliability crisis of reference-free metrics for grammatical error correction.** In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24913–24926, Suzhou, China. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025c. **Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human?** In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. Association for Computational Linguistics.
- Vansh Gupta, Sankalan Pal Chowdhury, Vilém Zouhar, Donya Roeein, and Mrinmaya Sachan. 2025. **Are large language models for education reliable across languages?** In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 612–631, Vienna, Austria. Association for Computational Linguistics.
- Abigail Gurin Schleifer, Beata Beigman Klebanov, Moriah Ariely, and Giora Alexandron. 2024. **Anna karenina strikes again: Pre-trained LLM embeddings may favor high-performing learners.** In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 391–402, Mexico City, Mexico. Association for Computational Linguistics.
- Ben Hamner, Jaison Morgan, lynnvande, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring. <https://kaggle.com/competitions/asap-aes>. Kaggle.
- Junzhi Han and Jinho D. Choi. 2025. **Beyond linear digital reading: An LLM-powered concept mapping approach for reducing cognitive load.** In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 805–817, Vienna, Austria. Association for Computational Linguistics.
- Luke Harding. 2025. **Utopian and dystopian visions: Steering a course for the responsible use of artificial intelligence (ai) in language testing and assessment.** *Language Testing*, 42(4):561–575.
- Ahatsham Hayat, Bilal Khan, and Mohammad Hasan. 2024. **Improving transfer learning for early forecasting of academic performance by contextualizing language models.** In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 137–148, Mexico City, Mexico. Association for Computational Linguistics.
- Junyi He and Xia Li. 2024. **Zero-shot cross-lingual automated essay scoring.** In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17819–17832, Torino, Italia. ELRA and ICCL.
- Angga Hidayat and Pipit Firmanti. 2024. **Navigating the tech frontier: a systematic review of technology integration in mathematics education.** *Cogent Education*, 11(1):2373559.
- Nils Hjortnaes, Daniel Dakota, Sandra Kübler, and Francis Tyers. 2024. **Evaluating automatic pronunciation scoring with crowd-sourced speech corpus annotations.** In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 67–77, Rennes, France. LiU Electronic Press.
- Wayne Holmes, Kaska Porayska-Pomsta, Ken Holstein, Emma Sutherland, Toby Baker, Simon Shum, Olga C. Santos, Mercedes Rodrigo, Mutlu Cukurova, Ig Bitencourt, and Kenneth Koedinger. 2022. **Ethics of**

- AI in Education: Towards a community-wide framework. *International Journal of Artificial Intelligence in Education*, 32:504–526.
- Anna Hülsing and Andrea Horbach. 2024. [Opinions are buildings: Metaphors in secondary education foreign language learning](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 78–95, Rennes, France. LiU Electronic Press.
- Leo Huovinen and Mika Härmäläinen. 2025. [LLM-assisted, iterative curriculum writing: A human-centered AI approach in Finnish higher education](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1002–1010, Vienna, Austria. Association for Computational Linguistics.
- Fareya Ikram, Alexander Scarlatos, and Andrew Lan. 2025. [Exploring LLMs for predicting tutor strategy and student outcomes in dialogues](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 765–779, Vienna, Austria. Association for Computational Linguistics.
- Michael Ilagan, Beata Beigman Klebanov, and Jamie Mikeska. 2024. [Automated evaluation of teacher encouragement of student-to-student interactions in a simulated classroom discussion](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 182–198, Mexico City, Mexico. Association for Computational Linguistics.
- Zhuoxuan Jiang, Tianyang Zhang, Peiyan Peng, Jing Chen, Yinong Xun, Haotian Zhang, Lichi Li, Yong Li, and Shaohua Zhang. 2025. [Towards generating controllable and solvable geometry problem by leveraging symbolic deduction engine](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1378–1398, Vienna, Austria. Association for Computational Linguistics.
- Ahmed Karim, Qiao Wang, and Zheng Yuan. 2025. [Beyond the score: Uncertainty-calibrated LLMs for automated essay assessment](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19631–19636, Suzhou, China. Association for Computational Linguistics.
- Anisia Katinskaia, Anh-Duc Vu, Jue Hou, Ulla Vanhatalo, Yiheng Wu, and Roman Yangarber. 2025. [Estimation of text difficulty in the context of language learning](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 594–611, Vienna, Austria. Association for Computational Linguistics.
- Anisia Katinskaia and Roman Yangarber. 2024. [GPT-3.5 for grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- Abdelhak Kelious, Mathieu Constant, and Christophe Coeur. 2024a. [Complex word identification: A comparative study between ChatGPT and a dedicated model for this task](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3645–3653, Torino, Italia. ELRA and ICCL.
- Abdelhak Kelious, Mathieu Constant, and Christophe Coeur. 2024b. [Investigating strategies for lexical complexity prediction in a multilingual setting using generative language models and supervised approaches](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 96–114, Rennes, France. LiU Electronic Press.
- Deokgi Kim, Joonyoung Jo, Byung-Won On, and Ingyu Lee. 2025a. [Representation-to-creativity \(R2C\): Automated holistic scoring model for essay creativity](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5272–5290, Albuquerque, New Mexico. Association for Computational Linguistics.
- Euigyum Kim, Seewoo Li, Salah Khalil, and Hyo Jeong Shin. 2025b. [STAIR-AIG: Optimizing the automated item generation process through human-AI collaboration for critical thinking assessment](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 920–930, Vienna, Austria. Association for Computational Linguistics.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Large language models are state-of-the-art evaluator for grammatical error correction](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.
- Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. [Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1011–1033, Vienna, Austria. Association for Computational Linguistics.
- Zahra Kolagar, Frank Zalkow, and Alessandra Zarcone. 2025. [Investigating methods for mapping learning objectives to bloom’s revised taxonomy in course descriptions for higher education](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 415–445, Vienna, Austria. Association for Computational Linguistics.

- Seonmin Koo, Jinsung Kim, Chanjun Park, and Heuiseok Lim. 2024. [Search if you don't know! knowledge-augmented Korean grammatical error correction with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 96–125, Miami, Florida, USA. Association for Computational Linguistics.
- Georgi Kostov. 2026. [The role of the parents in modern education: Educational policy and interaction with the family environment](#). *International Journal of Didactical Studies*, 7.
- Charles Koutcheme, Nicola Dainese, and Arto Hellas. 2024. [Using program repair as a proxy for language models' feedback ability in programming education](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 165–181, Mexico City, Mexico. Association for Computational Linguistics.
- Charles Koutcheme, Nicola Dainese, and Arto Hellas. 2025. [Direct repair optimization: Training small language models for educational program repair improves feedback](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 564–581, Vienna, Austria. Association for Computational Linguistics.
- Aomi Koyama, Masato Mita, Su-Youn Yoon, Yasufumi Takama, and Mamoru Komachi. 2025. [Targeted syntactic evaluation for grammatical error correction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21108–21125, Vienna, Austria. Association for Computational Linguistics.
- Klaus Krippendorff. 2011. [Computing krippendorff's alpha-reliability](#).
- Andrei Kucharavy, Cyril Vallez, and Dimitri Percia David. 2025. [LLMs protégés: Tutoring LLMs with knowledge gaps improves student learning outcome](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 248–257, Vienna, Austria. Association for Computational Linguistics.
- Aayush Kucheria, Nitin Sawhney, and Arto Hellas. 2025. [Comparing behavioral patterns of LLM and human tutors: A population-level analysis with the CIMA dataset](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 873–881, Vienna, Austria. Association for Computational Linguistics.
- Alexander Kwako and Christopher Ormerod. 2024. [Can language models guess your identity? analyzing demographic biases in AI essay scoring](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 78–86, Mexico City, Mexico. Association for Computational Linguistics.
- Mihwa Lee, Björn Rudzewitz, and Xiaobin Chen. 2024a. [Developing a pedagogically oriented interactive reading tool with teachers in the loops](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 115–125, Rennes, France. LiU Electronic Press.
- Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024b. [Unleashing large language models' proficiency in zero-shot essay scoring](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198, Miami, Florida, USA. Association for Computational Linguistics.
- Bernardo Leite and Henrique Lopes Cardoso. 2025. [Advancing question generation with joint narrative and difficulty control](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 647–659, Vienna, Austria. Association for Computational Linguistics.
- Shengjie Li and Vincent Ng. 2024. [Automated essay scoring: A reflection on the state of the art](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.
- Shengjie Li and Vincent Ng. 2025. [Graph-based multi-trait essay scoring](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33325–33351, Suzhou, China. Association for Computational Linguistics.
- Tianwen Li, Zhexiong Liu, Lindsay Matsumura, Elaine Wang, Diane Litman, and Richard Correnti. 2024. [Using large language models to assess young students' writing revisions](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 365–380, Mexico City, Mexico. Association for Computational Linguistics.
- Wei Li, Wen Luo, Guangyue Peng, and Houfeng Wang. 2025. [Explanation based in-context demonstrations retrieval for multilingual grammatical error correction](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4881–4897, Albuquerque, New Mexico. Association for Computational Linguistics.
- Ho Hung Lim and John Lee. 2024. [Improving readability assessment with ordinal log-loss](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 343–350, Mexico City, Mexico. Association for Computational Linguistics.
- Zhengyuan Liu, Stella Xin Yin, Dion Hoe-Lian Goh, and Nancy Chen. 2025. [COGENT: A curriculum-oriented framework for generating grade-appropriate](#)

- educational content. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 129–143, Vienna, Austria. Association for Computational Linguistics.
- Sarah Löber, Björn Rudzewitz, Daniela Verratti Souto, Luisa Ribeiro-Flucht, and Xiaobin Chen. 2024. [Developing a web-based intelligent language assessment platform powered by natural language processing technologies](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 126–136, Rennes, France. LiU Electronic Press.
- Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024. [No error left behind: Multilingual grammatical error correction with pre-trained translation models](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1222, St. Julian’s, Malta. Association for Computational Linguistics.
- Wanjing (Anyu) Ma, Michael Flor, and Zuowei Wang. 2025. [Automatic generation of inference making questions for reading comprehension assessments](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 398–414, Vienna, Austria. Association for Computational Linguistics.
- Zhenjiang Mao, Artem Bislouk, Rohith Nama, and Ivan Ruchkin. 2025. [Temporalizing confidence: Evaluation of chain-of-thought reasoning with signal temporal logic](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 882–890, Vienna, Austria. Association for Computational Linguistics.
- Jacek Marciniak, Marek Kubis, Michał Gulczyński, Adam Szpilkowski, Adam Wieczarek, and Marcin Szczepański. 2025. [Improving AI assistants embedded in short e-learning courses with limited textual content](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 794–804, Vienna, Austria. Association for Computational Linguistics.
- Daria Martynova, Jakub Macina, Nico Daheim, Nilay Yalcin, Xiaoyu Zhang, and Mrinmaya Sachan. 2025. [Can LLMs effectively simulate human learners? teachers’ insights from tutoring LLM students](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 100–117, Vienna, Austria. Association for Computational Linguistics.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025. [The MultiGEC-2025 shared task on multilingual grammatical error correction at NLP4CALL](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 1–33, Tallinn, Estonia. University of Tartu Library.
- Noah-Manuel Michael and Andrea Horbach. 2025. [GermDetect: Verb placement error detection datasets for learners of Germanic languages](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 818–829, Vienna, Austria. Association for Computational Linguistics.
- Marvin Minsky. 1974. [A framework for representing knowledge](#). *MIT Artificial Intelligence Laboratory Memo*, 306.
- Adriana Mirabella and Dominique Brunato. 2025. [Exploring LLM-based assessment of Italian middle school writing: A pilot study](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 708–715, Vienna, Austria. Association for Computational Linguistics.
- Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. [Towards automated document revision: Grammatical error correction, fluency edits, and beyond](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.
- Rina Miyata, Toru Urakawa, Hideaki Tamori, and Tomoyuki Kajiwaru. 2025. [Unsupervised sentence readability estimation based on parallel corpora for text simplification](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 499–504, Vienna, Austria. Association for Computational Linguistics.
- Phoebe Mulcaire and Nitin Madnani. 2025. [Span labeling with large language models: Shell vs. meat](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 850–859, Vienna, Austria. Association for Computational Linguistics.
- Ricardo Muñoz Sánchez, David Alfter, Simon Dobnik, Maria Irena Szawerna, and Elena Volodina. 2024a. [Jingle BERT, frozen all the way: Freezing layers to identify CEFR levels of second language learners using BERT](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 137–152, Rennes, France. LiU Electronic Press.
- Ricardo Muñoz Sánchez, Simon Dobnik, and Elena Volodina. 2024b. [Harnessing GPT to study second language learner essays: Can we use perplexity to determine linguistic competence?](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 414–427, Mexico City, Mexico. Association for Computational Linguistics.

- Karthika N J, Krishnakant Bhatt, Ganesh Ramakrishnan, and Preethi Jyothi. 2025. [LEVOS: Leveraging vocabulary overlap with Sanskrit to generate technical lexicons in Indian languages](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 258–265, Vienna, Austria. Association for Computational Linguistics.
- Kamel Nebhi, Amrita Panesar, and Hans Bantilan. 2025. [End-to-end automated item generation and scoring for adaptive English writing assessment with large language models](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 968–977, Vienna, Austria. Association for Computational Linguistics.
- Allen Newell, J. C. Shaw, and Herbert Simon. 1958. [Elements of a theory of human problem solving](#). *Psychological Review*, 65:151–166.
- 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.
- Iglikla Nikolova-Stoupak, Serge Bibauw, Amandine Dumont, Françoise Stas, Patrick Watrin, and Thomas François. 2024. [Generating contexts for ESP vocabulary exercises with LLMs](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 153–175, Rennes, France. LiU Electronic Press.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Kornienko, and Igor Samokhin. 2024. [Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.
- OpenAI. 2026. ChatGPT. <https://chat.openai.com>. Accessed: May 13, 2026.
- Robert Östling, Murathan Kurfali, and Andrew Caines. 2025. [LLM-based post-editing as reference-free GEC evaluation](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 213–224, Vienna, Austria. Association for Computational Linguistics.
- Austin Pack, Alex Barrett, and Juan Escalante. 2024. [Large language models and automated essay scoring of English language learner writing: Insights into validity and reliability](#). *Computers and Education: Artificial Intelligence*, 6:100234.
- Benjamin Paddags, Daniel Hershcovich, and Valkyrie Savage. 2024. [Automated sentence generation for a spaced repetition software](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 351–364, Mexico City, Mexico. Association for Computational Linguistics.
- Francesca Padovani, Caterina Marchesi, Eleonora Pasqua, Martina Galletti, and Daniele Nardi. 2024. [Automatic text simplification: A comparative study in Italian for children with language disorders](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 176–186, Rennes, France. LiU Electronic Press.
- Sankalan Pal Chowdhury, Nico Daheim, Ekaterina Kochmar, Jakub Macina, Donya Rooein, Mrinmaya Sachan, and Shashank Sonkar. 2025a. [Large language models for education: Understanding the needs of stakeholders, current capabilities and the path forward](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1–10, Vienna, Austria. Association for Computational Linguistics.
- Sankalan Pal Chowdhury, Terry Jingchen Zhang, Donya Rooein, Dirk Hovy, Tanja Käser, and Mrinmaya Sachan. 2025b. [Educators’ perceptions of large language models as tutors: Comparing human and AI tutors in a blind text-only setting](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 356–374, Vienna, Austria. Association for Computational Linguistics.
- Seymour Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Harvester Press.
- Nisarg Parikh, Alexander Scarlatos, Nigel Fernandez, Simon Woodhead, and Andrew Lan. 2025. [LookA-like: Consistent distractor generation in math MCQs](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 294–311, Vienna, Austria. Association for Computational Linguistics.
- Geon Park, Jiwoo Song, Gihyeon Choi, Juoh Sun, and Harksoo Kim. 2025. [K-NLPers at BEA 2025 shared task: Evaluating the quality of AI tutor responses with GPT-4.1](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1145–1163, Vienna, Austria. Association for Computational Linguistics.
- Juan Antonio Pérez-Ortiz, Miquel Esplà-Gomis, Víctor M. Sánchez-Cartagena, Felipe Sánchez-Martínez, Roman Chernysh, Gabriel Mora-Rodríguez, and Lev Berezhnoy. 2024. [A conversational intelligent tutoring system for improving English proficiency of non-native speakers via debriefing of online meeting transcriptions](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 187–198, Rennes, France. LiU Electronic Press.

- Kseniia Petukhova and Ekaterina Kochmar. 2025. [Intent matters: Enhancing AI tutoring with fine-grained pedagogical intent annotation](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 860–872, Vienna, Austria. Association for Computational Linguistics.
- Yin Poon, Qiong Wang, John S. Y. Lee, Yu Yan Lam, and Samuel Kai Wah Chu. 2025. [PIRLS category-specific question generation for reading comprehension](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 72–80, Tallinn, Estonia. University of Tartu Library.
- Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Jungyeul Park. 2025. [Multilingual grammatical error annotation: Combining language-agnostic framework with language-specific flexibility](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 202–212, Vienna, Austria. Association for Computational Linguistics.
- Muhammad Reza Qorib, Alham Fikri Aji, and Hwee Tou Ng. 2024. [Efficient and interpretable grammatical error correction with mixture of experts](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17127–17138, Miami, Florida, USA. Association for Computational Linguistics.
- Chatrine Qwaider, Bashar Alhafni, Kirill Chirkunov, Nizar Habash, and Ted Briscoe. 2025. [Enhancing Arabic automated essay scoring with synthetic data and error injection](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 549–563, Vienna, Austria. Association for Computational Linguistics.
- Saed Rezayi, Le An Ha, Yiyun Zhou, Andrew Houriet, Angelo D’Addario, Peter Baldwin, Polina Harik, Ann King, and Victoria Yaneva. 2025. [Automated scoring of communication skills in physician-patient interaction: Balancing performance and scalability](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 891–897, Vienna, Austria. Association for Computational Linguistics.
- Luisa Ribeiro-Flucht, Xiaobin Chen, and Detmar Meurers. 2024. [Explainable AI in language learning: Linking empirical evidence and theoretical concepts in proficiency and readability modeling of Portuguese](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 199–209, Mexico City, Mexico. Association for Computational Linguistics.
- Luisa Ribeiro-Flucht, Xiaobin Chen, and Detmar Meurers. 2025. [A framework for proficiency-aligned grammar practice in LLM-based dialogue systems](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 978–987, Vienna, Austria. Association for Computational Linguistics.
- David J. Roach  . 2017. *Intercoder Reliability Techniques: Percent Agreement*. SAGE Publications, Inc, California.
- Donya Rooein, Paul R  ttger, Anastassia Shaitarova, and Dirk Hovy. 2024. [Beyond flesch-kincaid: Prompt-based metrics improve difficulty classification of educational texts](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 54–67, Mexico City, Mexico. Association for Computational Linguistics.
- Alla Rozovskaya. 2024. [Universal Dependencies for learner Russian](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17112–17119, Torino, Italia. ELRA and ICCL.
- Annabella Sakunkoo and Jonathan Sakunkoo. 2025. [Name of thrones: How do LLMs rank student names in status hierarchies based on race and gender?](#) In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 697–707, Vienna, Austria. Association for Computational Linguistics.
- Ignacio Sastre, Leandro Alfonso, Facundo Fleitas, Federico Gil, Andr  s Lucas, Tom  s Spoturno, Santiago G  ngora, Aiala Ros  , 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)*, pages 618–626, Mexico City, Mexico. Association for Computational Linguistics.
- Andreas S  uberli, Diego Frassinelli, and Barbara Plank. 2025. [Do LLMs give psychometrically plausible responses in educational assessments?](#) In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 266–278, Vienna, Austria. Association for Computational Linguistics.
- Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024. [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?](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 1–10, Mexico City, Mexico. Association for Computational Linguistics.
- Alexander Scarlatos, Wanyong Feng, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. [Improving automated distractor generation for math multiple-choice questions with overgenerate-and-rank](#). In *Proceedings of the 19th Workshop on Innovative Use*

- of NLP for Building Educational Applications (BEA 2024), pages 222–231, Mexico City, Mexico. Association for Computational Linguistics.
- Nils-Jonathan Schaller, Yuning Ding, Andrea Horbach, Jennifer Meyer, and Thorben Jansen. 2024. [Fairness in automated essay scoring: A comparative analysis of algorithms on German learner essays from secondary education](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 210–221, Mexico City, Mexico. Association for Computational Linguistics.
- Nils-Jonathan Schaller, Yuning Ding, Thorben Jansen, and Andrea Horbach. 2025. [Don't score too early! evaluating argument mining models on incomplete essays](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 345–355, Vienna, Austria. Association for Computational Linguistics.
- Veronica Schmalz and Anaïs Tack. 2025. [Can GPTZero's AI vocabulary distinguish between LLM-generated and student-written essays?](#) In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 937–952, Vienna, Austria. Association for Computational Linguistics.
- Olga Seminck, Yoann Dupont, Mathieu Dehouck, Qi Wang, Noé Durandard, and Margo Novikov. 2025. [Lattice @MultiGEC-2025: A spiteful multilingual language error correction system using LLaMA](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 34–41, Tallinn, Estonia. University of Tartu Library.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Sagion. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Mayank Sharma and Jason Zhang. 2025. [Decoding actionability: A computational analysis of teacher observation feedback](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 898–907, Vienna, Austria. Association for Computational Linguistics.
- Kevin Shi and Karttikeya Mangalam. 2025. [UPSC2M: Benchmarking adaptive learning from two million MCQ attempts](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 931–936, Vienna, Austria. Association for Computational Linguistics.
- Takumi Shibata and Yuichi Miyamura. 2025. [LCES: Zero-shot automated essay scoring via pairwise comparisons using large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29988–30001, Suzhou, China. Association for Computational Linguistics.
- Mariana Shimabukuro, Deval Panchal, and Christopher Collins. 2025. [LangEye: Toward 'anytime' learner-driven vocabulary learning from real-world objects](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 446–459, Vienna, Austria. Association for Computational Linguistics.
- Astha Singh, Mark Torrance, and Evgeny Chukharev. 2025. [EyeLLM: Using lookback fixations to enhance human-LLM alignment for text completion](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 841–849, Vienna, Austria. Association for Computational Linguistics.
- Li Siyan, Teresa Shao, Julia Hirschberg, and Zhou Yu. 2024. [Using adaptive empathetic responses for teaching English](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 34–53, Mexico City, Mexico. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Alexey Sorokin and Regina Nasyrova. 2025. [LLMs in alliance with edit-based models: advancing in-context learning for grammatical error correction by specific example selection](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 517–534, Vienna, Austria. Association for Computational Linguistics.
- KV Aditya Srivatsa, Kaushal Maurya, and Ekaterina Kochmar. 2025a. [Can LLMs reliably simulate real students' abilities in mathematics and reading comprehension?](#) In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 988–1001, Vienna, Austria. Association for Computational Linguistics.
- Kv Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2025b. [LLMs cannot spot math errors, even when allowed to peek into the solution](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages

- 10914–10928, Suzhou, China. Association for Computational Linguistics.
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM prompting strategies for joint essay scoring and feedback generation](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2024. [Synthetic data generation for low-resource grammatical error correction with tagged corruption models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 11–16, Mexico City, Mexico. Association for Computational Linguistics.
- Ryszard Staruch. 2025. [UAM-CSI at MultiGEC-2025: Parameter-efficient LLM fine-tuning for multilingual grammatical error correction](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 42–49, Tallinn, Estonia. University of Tartu Library.
- Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. 2025. [Adapting LLMs for minimal-edit grammatical error correction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 118–128, Vienna, Austria. Association for Computational Linguistics.
- Bernardo Stearns, Nicolas Ballier, Thomas Gaillat, Andrew Simpkin, and John P. McCrae. 2024. [Evaluating the generalisation of an artificial learner](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 199–208, Rennes, France. LiU Electronic Press.
- Kevin Stowe, Benny Longwill, Alyssa Francis, Tatsuya Aoyama, Debanjan Ghosh, and Swapna Somasundaran. 2024. [Identifying fairness issues in automatically generated testing content](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 232–250, Mexico City, Mexico. Association for Computational Linguistics.
- David Strohmaier and Paula Buttery. 2024. [Semantic error prediction: Estimating word production complexity](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 209–225, Rennes, France. LiU Electronic Press.
- Jiamin Su, Yibo Yan, Fangteng Fu, Zhang Han, Jingheng Ye, Xiang Liu, Jiahao Huo, Huiyu Zhou, and Xuming Hu. 2025. [EssayJudge: A multi-granular benchmark for assessing automated essay scoring capabilities of multimodal large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6363–6389, Vienna, Austria. Association for Computational Linguistics.
- Hakyung Sung, Karla Csuros, and Min-Chang Sung. 2025. [Comparing human and LLM proofreading in L2 writing: Impact on lexical and syntactic features](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 11–23, Vienna, Austria. Association for Computational Linguistics.
- Harini Suresh and John Gutttag. 2021. [A framework for understanding sources of harm throughout the machine learning life cycle](#). In *Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO)*.
- 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)*, pages 635–639, Mexico City, Mexico. Association for Computational Linguistics.
- 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)*, pages 512–521, Mexico City, Mexico. Association for Computational Linguistics.
- Chenming Tang, Fanyi Qu, and Yunfang Wu. 2024. [Ungrammatical-syntax-based in-context example selection for grammatical error correction](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1758–1770, Mexico City, Mexico. Association for Computational Linguistics.
- Rajneesh Tiwari and Pranshu Rastogi. 2025. [Phaedrus at BEA 2025 shared task: Assessment of mathematical tutoring dialogues through tutor identity classification and actionability evaluation](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1098–1107, Vienna, Austria. Association for Computational Linguistics.
- Guillaume Toussaint, Yannick Parmentier, and Claire Gardent. 2024. [GRAMEX: Generating controlled grammar exercises from various sources](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 226–234, Rennes, France. LiU Electronic Press.
- Nhat Tran, Diane Litman, Benjamin Pierce, Richard Correnti, and Lindsay Clare Matsumura. 2025. [Improving in-context learning example retrieval for classroom discussion assessment with re-ranking and](#)

- label ratio regulation. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 752–764, Vienna, Austria. Association for Computational Linguistics.
- Gladys Tyen, Andrew Caines, and Paula Buttery. 2024. *LLM chatbots as a language practice tool: a user study*. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 235–247, Rennes, France. LiU Electronic Press.
- Till Überrück-Fries, Agata Savary, and Agnieszka Dryjańska. 2024. *Sailing through multiword expression identification with Wiktionary and linguse: A case study of language learning*. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 248–262, Rennes, France. LiU Electronic Press.
- Ahmet Yavuz Uluslu and Gerold Schneider. 2025. *Investigating linguistic abilities of LLMs for native language identification*. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 81–88, Tallinn, Estonia. University of Tartu Library.
- UNESCO. 2026. *Global report on teachers: Addressing teacher shortages and transforming the profession*.
- Felipe Urrutia, Cristian Buc, Roberto Araya, and Valentin Barriere. 2025. *Unsupervised automatic short answer grading and essay scoring: A weakly supervised explainable approach*. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 38–54, Vienna, Austria. Association for Computational Linguistics.
- Martin Vainikko, Taavi Kamarik, Karina Kert, Krista Liin, Silvia Maine, Kais Allkivi, Annekatrin Kaivapalu, and Mark Fishel. 2025. *Paragraph-level error correction and explanation generation: Case study for Estonian*. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 953–967, Vienna, Austria. Association for Computational Linguistics.
- Hariram Veeramani, Surendrabikram Thapa, Natarajan Balaji Shankar, and Abeer Alwan. 2024. *Large language model-based pipeline for item difficulty and response time estimation for educational assessments*. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 561–566, Mexico City, Mexico. Association for Computational Linguistics.
- Georgios Velentzas, Andrew Caines, Rita Borgo, Erin Pacquetet, Clive Hamilton, Taylor Arnold, Diane Nicholls, Paula Buttery, Thomas Gaillat, Nicolas Ballier, and Helen Yannakoudakis. 2024. *Logging keystrokes in writing by English learners*. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10725–10746, Torino, Italia. ELRA and ICCL.
- Anh-Duc Vu, Jue Hou, Anisia Katinskaia, Ching-Fan Sheu, and Roman Yangarber. 2025. *A Bayesian approach to inferring prerequisite structures and topic difficulty in language learning*. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 737–751, Vienna, Austria. Association for Computational Linguistics.
- Deliang Wang, Chao Yang, and Gaowei Chen. 2025a. *Wonderland_EDU@HKU at BEA 2025 shared task: Fine-tuning large language models to evaluate the pedagogical ability of AI-powered tutors*. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1040–1048, Vienna, Austria. Association for Computational Linguistics.
- Jian Wang, Yinpei Dai, Yichi Zhang, Ziqiao Ma, Wenjie Li, and Joyce Chai. 2025b. *Training turn-by-turn verifiers for dialogue tutoring agents: The curious case of LLMs as your coding tutors*. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12416–12436, Vienna, Austria. Association for Computational Linguistics.
- Junling Wang, Anna Rutkiewicz, April Wang, and Mrinmaya Sachan. 2025c. *Generating pedagogically meaningful visuals for math word problems: A new benchmark and analysis of text-to-image models*. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11229–11257, Vienna, Austria. Association for Computational Linguistics.
- Xiaoman Wang, Dan Yuan, Xin Liu, Yike Zhao, Xiaoxiao Zhang, Xizhi Chen, and Yunshi Lan. 2025d. *VisCGEC: Benchmarking the visual Chinese grammatical error correction*. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5054–5068, Albuquerque, New Mexico. Association for Computational Linguistics.
- Xiaoying Wang, Lingling Mu, Jingyi Zhang, and Hongfei Xu. 2024a. *Multi-pass decoding for grammatical error correction*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9916, Miami, Florida, USA. Association for Computational Linguistics.
- Yixuan Wang, Baoxin Wang, Yijun Liu, Dayong Wu, and Wanxiang Che. 2024b. *LM-combiner: A contextual rewriting model for Chinese grammatical error correction*. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10675–10685, Torino, Italia. ELRA and ICCL.
- Yupei Wang, Renfen Hu, and Zhe Zhao. 2024c. *Beyond agreement: Diagnosing the rationale alignment of automated essay scoring methods based on*

- linguistically-informed counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8906–8925, Miami, Florida, USA. Association for Computational Linguistics.
- World Inequality Lab. 2026. [World inequality report 2026](#). [Accessed 31-03-2026].
- Tianxiang Wu, Han Chen, Luozheng Qin, Ziqiang Cao, and Chunhui Ai. 2024. [Improving copy-oriented text generation via EDU copy mechanism](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8768–8780, Torino, Italia. ELRA and ICCL.
- Yuyang Yan, Hui Liu, and Toby Chau. 2025. [A systematic review of AI ethics in education: Challenges, policy gaps, and future directions](#). *Journal of Global Information Management*, 33(1):1–50.
- Kevin P. Yancey, Andrew Runge, Geoffrey LaFlair, and Phoebe Mulcaire. 2024. [BERT-IRT: Accelerating item piloting with BERT embeddings and explainable IRT models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 428–438, Mexico City, Mexico. Association for Computational Linguistics.
- Victoria Yaneva, Kai North, Peter Baldwin, Le An Ha, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024a. [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)*, pages 470–482, Mexico City, Mexico. Association for Computational Linguistics.
- Victoria Yaneva, King Yiu Suen, Le An Ha, Janet Mee, Milton Quranda, and Polina Harik. 2024b. [Automated scoring of clinical patient notes: Findings from the Kaggle competition and their translation into practice](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 87–98, Mexico City, Mexico. Association for Computational Linguistics.
- Haihui Yang and Xiaojun Quan. 2024. [Alirector: Alignment-enhanced Chinese grammatical error corrector](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2531–2546, Bangkok, Thailand. Association for Computational Linguistics.
- Haiyin Yang, Zoey Liu, and Stefanie Wulff. 2025. [Using NLI to identify potential collocation transfer in L2 English](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 687–696, Vienna, Austria. Association for Computational Linguistics.
- Sahar Yarmohammadtoosky, Yiyun Zhou, Victoria Yaneva, Peter Baldwin, Saed Rezayi, Brian Clauser, and Polina Harik. 2025. [Enhancing security and strengthening defenses in automated short-answer grading systems](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 830–840, Vienna, Austria. Association for Computational Linguistics.
- Mazen Yasser, Mariam Saeed, Hossam Elkordi, and Ayman Khalafallah. 2025. [Averroes at BEA 2025 shared task: Verifying mistake identification in tutor, student dialogue](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1121–1126, Vienna, Austria. Association for Computational Linguistics.
- Jingheng Ye, Zishan Xu, Yinghui Li, Linlin Song, Qingyu Zhou, Hai-Tao Zheng, Ying Shen, Wenhao Jiang, Hong-Gee Kim, Ruitong Liu, Xin Su, and Zifei Shan. 2025. [CLEME2.0: Towards interpretable evaluation by disentangling edits for grammatical error correction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 204–222, Vienna, Austria. Association for Computational Linguistics.
- Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. 2025. [DREsS: Dataset for rubric-based essay scoring on EFL writing](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13439–13454, Vienna, Austria. Association for Computational Linguistics.
- 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)*, pages 551–560, Mexico City, Mexico. Association for Computational Linguistics.
- Fabian Zehner, Hyo Jeong Shin, Emily Kerzabi, Andrea Horbach, Sebastian Gombert, Frank Goldhammer, Torsten Zesch, and Nico Andersen. 2025. [Down the cascades of omethi: Hierarchical automatic scoring in large-scale assessments](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 660–671, Vienna, Austria. Association for Computational Linguistics.
- Torsten Zesch, Dominic Gardner, and Marie Bexte. 2025. [Transformer-based real-word spelling error feedback with configurable confusion sets](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 375–383, Vienna, Austria. Association for Computational Linguistics.
- Yike Zhao, Simin Guo, Ziqing Yang, Shifan Han, Dahua Lin, and Fei Tan. 2025. [More data or better data? a critical analysis of data selection and synthesis for mathematical reasoning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 618–629,

Suzhou (China). Association for Computational Linguistics.

A A Structured Taxonomy for Ethical and Stakeholder Review of EduNLP Research

Figure 10¹² presents the complete taxonomy developed in this review, offered here as a standalone contribution. The taxonomy is organised around the three research questions and a concluding recommendations dimension. Beyond its role in this review, the taxonomy can be reused for future surveys of EdTech research. It can also serve as a practical self-audit tool: researchers can use it to situate themselves within the EduNLP space, and assess the rigour and inclusivity of their work before submission.

B ACL Anthology Main Conference Search

We retrieve all papers with at least one of the following search terms in the title or abstract. The venues included are: ACL, EACL, NAACL, EMNLP, LREC-COLING, and Findings, for the years 2024 and 2025. The search terms were developed through internal discussion and discussion with other researchers in the EduNLP field. The search terms are as follows:

- “automated essay scoring”,
- “automated writing evaluation”,
- “short answer grading”,
- “automatic short answer grading”,
- “open-ended response assessment”,
- “automated assessment of spoken responses”,
- “spoken response scoring”,
- “speech-based assessment”,
- “automatic speech scoring”,
- “dialogue-based tutoring”,
- “spoken dialogue system education”,
- “intelligent tutoring systems NLP”,
- “student modeling”,
- “learner modeling”,
- “knowledge tracing”,
- “learner cognition modeling”,
- “educational data mining NLP”,
- “learning analytics text”,
- “game-based learning assessment”,
- “stealth assessment”,
- “peer assessment NLP”,

- “peer review automated feedback”,
- “automated feedback generation”,
- “formative feedback writing”,
- “grammatical error correction”,
- “grammar error detection”,
- “lexical complexity prediction”,
- “text simplification for learners”,
- “multimodal learning analytics”,
- “generative AI in education”,
- “mathematic education”,
- “math education”,
- “math word problems”,
- “mathematical reasoning”,
- “student error in mathematics”,
- “intelligent tutoring system math”,
- “knowledge tracing mathematics”,
- “misconception detection mathematics”.

C Extraction Schema

For extracting the entities relevant to our research questions, we used the following schema:

- [RQ2] Author affiliations
- [RQ1] Specific task worked on
- [RQ1] Datasets used and availability
- [RQ1] Explicit motivation for the paper and associated quotes
- [RQ2] Stakeholders mentioned (multi-label; the options being *Learner/student*, *Teacher*, *School/university*, *Paper author*, *Other researcher*, *Domain expert*, *Parent*, *Governmental body*, *Industry*, *Non-profit* which includes large-standardised testing providers, and *None / N.A.*), and associated quotes
- [RQ2] Stakeholders included in the research (multi-label; same list as above), as well as their respective level of inclusion (multi-label with *High*, *Middling* and *Low*) and associated quotes for how they are included
- [RQ1] Context in which the system deployed (if any) and relevant quotes
- [RQ2] Explicit stakeholder incentives
- [RQ2] Implicit stakeholder incentives
- [RQ3] Risk, concerns and limitations raised, associated quotes, level of engagement (multi-label with *High*, *Middling* and *Low*) and measures taken to address risk

¹²This figure was created in app.xmind.com.

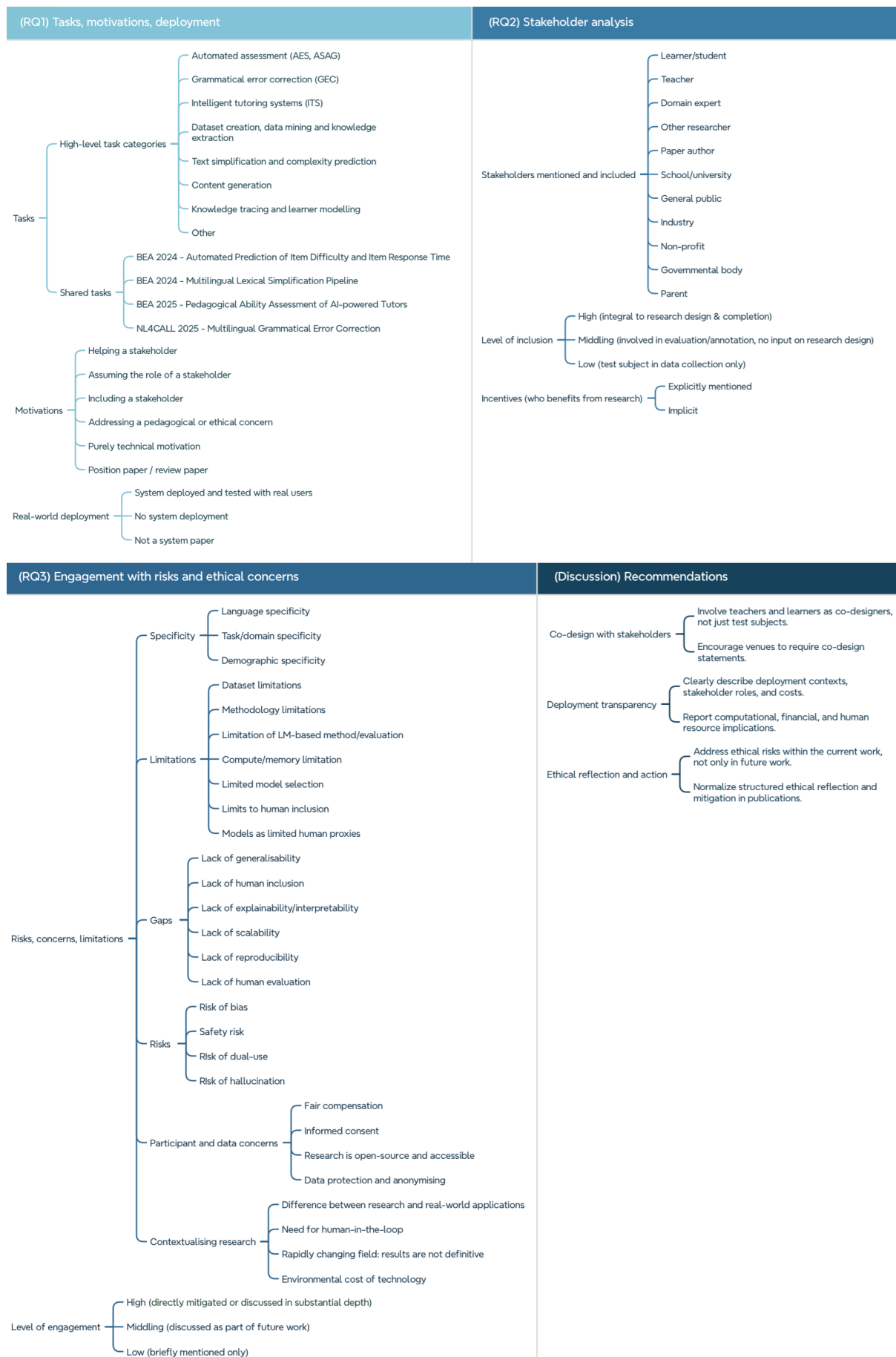


Figure 10: Detailed taxonomy for reviewing EduNLP research.

- **[RQ3]** Future directions/aspirations mentioned and relevant quotes
- **[RQ1]** Entities acknowledged (including funding)

Once phase (3) was completed, the three annotators independently extracted high-level categories and labels from the data:

- **[RQ1]** Mapping specific tasks to high-level tasks (as reported in Figure 2)
- **[RQ2]** Mapping free-text dataset names to unique labels (as reported in Figure 3) and their availability (as reported in Figure 11)
- **[RQ1]** Categorising free-text motivations to high-level labels (as reported in Figure 12) and the mentioned stakeholders (as reported in Figure 4)
- **[RQ1]** Mapping context deployment to high-level labels (as reported in Figure 5)
- **[RQ2]** Extracting standardised stakeholder types from explicit and implicit incentives (as reported in Figure 9)
- **[RQ3]** Mapping risks, concerns and limitations to high-level categories (as reported in Figure 18) and the authors’ level of engagement associated to each (as reported in Figure 19)
- **[RQ3]** Mapping future directions and aspirations high-level categories (as reported in Figure 20)

For each dimension, the mapping was made by one of the three annotators independently. This was considered sufficient given that the high-level categories were derived directly from the extracted free-text data rather than applied to raw papers: the iterative reconciliation process in phases (1) and (2) had already established shared interpretive norms among annotators, and the categorisation task at this stage involved consolidating labels that were already grounded in agreed extractions rather than making independent judgements about unseen material.

D Agreement Computation

Table 1 reports agreement for the free-text dimensions. Table 6 includes examples of how

Dimension	Agreement
Task	0.95
Datasets used*	0.87
Dataset availability	0.93
Methods	0.93
Evaluation	0.92
Motivation	0.96
Deployment	1.0
Explicit incentives	0.72
Implicit incentives	0.53
Risks/concerns*	0.57
Measures taken to address risks/concerns*	0.91
Future directions*	0.69
Future deployment	0.92

Table 1: Agreement for free-text annotation dimensions. For dimensions with an asterisk, we computed per-paper percentage agreement. For the other dimensions, we computed majority percentages (i.e. 0 if none of the annotators agree, 0.67 if 2/3 annotators agree, and 1 if 3/3 annotators agree). The values reported correspond to the averages across all 25 papers in the shared batch. Computation examples are included in Table 6 and Table 7.

Label	PA	α
Overall	0.91	0.49
Domain experts	0.84	0.64
General user	0.97	0.74
Industry/company	0.97	0.49
Learners	0.92	-0.03
None / N.A.	0.95	-0.01
Other Researchers	0.79	0.44
Paper authors	0.92	0.46
Policy makers/governments/ministries	0.95	0.31
Schools/universities	0.79	0.27
Special needs/disability user	0.97	0.0
Teachers	0.89	0.79

Table 2: Average percentage agreement (PA) and inter-annotator agreement (Krippendorff’s α) for “stakeholders mentioned”.

percentage agreement (PA) was calculated for the following free-text dimensions: “datasets used”, “risks/concerns”, “measures taken to address risks/concerns”, and “future directions”. As these dimensions often included many different elements (e.g., dataset names for the “datasets used” dimension and specific suggestions for future research for “future directions”), PA was calculated in the form of pairwise agreements (at the level of the paper, with the score reported in Table 1 corre-

Label	PA	α
Overall	0.94	0.7
Domain experts	0.89	0.61
General user	1.0	1.0
Industry/company	0.97	0.49
Learners	0.92	0.71
None / N.A.	0.92	0.84
Other Researchers	0.92	0.53
Paper authors	0.92	0.46
Policy makers/governments/ministries	0.97	0.0
Schools/universities	0.97	0.0
Teachers	0.97	0.84

Table 3: Average percentage agreement (PA) and inter-annotator agreement (Krippendorff’s α) for “stakeholders included”.

Label	PA	α
Overall	0.88	0.61
High	0.84	0.51
Middling	0.84	0.41
Minimal	0.92	0.53
None / N.A.	0.92	0.84

Table 4: Average percentage agreement (PA) and inter-annotator agreement (Krippendorff’s alpha) for “level of inclusion stakeholders included”.

Label	PA	α
Overall	0.84	0.52
High	0.87	0.66
Middling	0.79	0.56
Minimal	0.79	0.57
None / N.A.	0.92	0.37

Table 5: Average percentage agreement (PA) and inter-annotator agreement (Krippendorff’s alpha) for “level of engagement risks/concerns”.

sponding to the mean of these per-paper pairwise agreement values).

Table 7 contains examples of how PA was calculated for the following free-text dimensions: “task”, “dataset availability”, “methods”, “evaluation”, “motivation”, “deployment”, “explicit incentives”, “implicit incentives”, and “future deployment”. As these dimensions virtually always included only a very limited number of core elements, PA was calculated in the form of majority percentages (at the level of the paper, with the score reported in Table 1 corresponding to the mean of

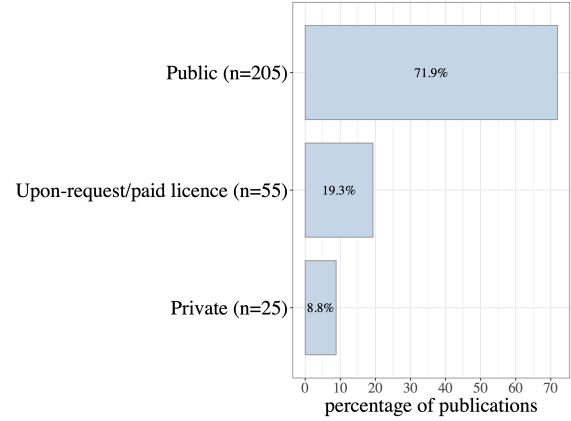


Figure 11: Availability of datasets used in the surveyed papers.

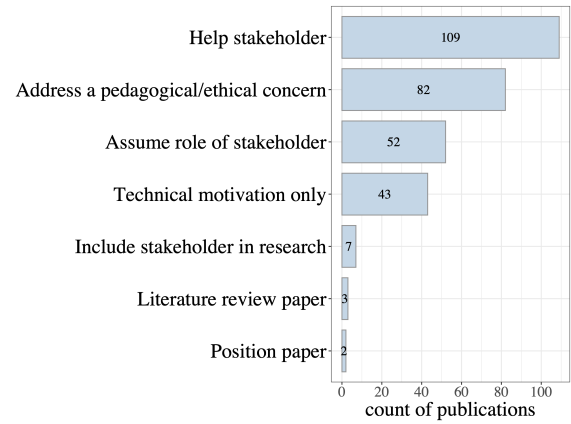


Figure 12: Distribution of papers’ explicit motivations for the task; papers may belong to more than one category.

these per-paper majority percentage values).

Tables 2 to 5 further present the agreement for the multilabel dimensions (stakeholders mentioned, stakeholders included, level of inclusion stakeholders included, and level of engagement risks/concerns).

E Mapping Papers to High-Level Tasks

The mapping of the analysed papers to the eight high-level tasks is presented in Table 8.

F Availability of Datasets

The availability of the datasets used in the analysed papers is presented in Figure 11.

G Explicit Motivations

The distribution of the papers’ explicit motivations for the research conducted is visualised in Figure 12.

Paper ID	Annotator 1	Annotator 2	Annotator 3	Computation
Datasets				
2025.bea-1.38	LORuGEC RULEC-GEC RU-LAng8 GERA	LORuGEC RULEC-GEC RU-LAng8 GERA	LORuGEC RULEC-GEC RU-LAng8 GERA English_BEA	• LORuGEC: 3/3 PWA • RULEC-GEC: 3/3 PWA • RU-LAng8: 3/3 PWA • GERA: 3/3 PWA • English_BEA: 1/3 PWA → PA paper = 13/15 = 86.87%
2025.bea-1.72	custom_dataset EKI-L2 Estonian National Corpus EstGEC-L2	custom_dataset EKI-L2 EstGEC-L2	EKI-L2	• custom_dataset: 1/3 PWA • EKI-L2: 3/3 PWA • Estonian National Corpus: 1/3 PWA • EstGEC-L2: 1/3 PWA → PA paper = 6/12 = 50%
Risks/concerns				
2025.acl-long.1026	Limited dataset size; Error Operation Ratio Limitation; limited language coverage; fair pay of annotators	Limited to English; ensured fair compensation for the annotators; limited number of minimal pairs due to manual creation	Manual annotation; dataset size; ratio of error operations is not controlled for; limited focus (on English)	• (Remunerate) annotators: 3/3 PWA • (Limited) language/resource coverage: 3/3 PWA • Error operation ratio: 1/3 PWA → PA paper = 7/9 = 77.78%

Table 6: Examples of percentage agreement (PA) computation for free-text annotation dimensions. “PWA” stands for pairwise agreement among the annotators (i.e. Annotator 1 compared to 2, Annotator 1 compared to 3, and Annotator 2 compared to 3). Note that *absence* of annotation also counts as agreement (e.g., between Annotator 1 and 2 for the “English_BEA” dataset).

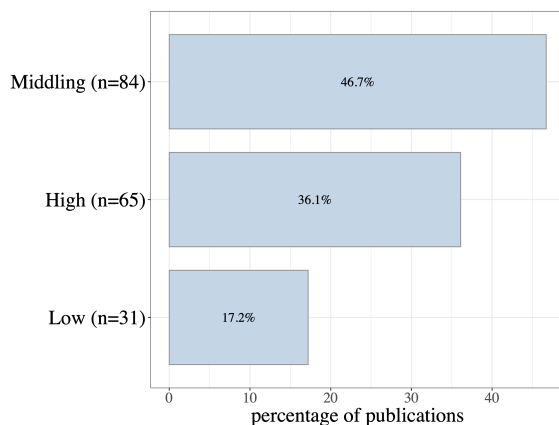


Figure 13: Overall level of inclusion of included stakeholders; we distinguish 3 levels: *High* (integral to research design & completion), *Middling* (involved in data evaluation or annotation, but have no input on research design), and *Low* (test subjects in data collection only).

H Stakeholders Included and Mentioned

As the authors are the primary stakeholders of the research, we present demographic information in terms of country of author affiliation in Figure 15. Secondly, Figure 13 visualises the level of inclusion of the stakeholders included in the research.

I Acknowledged Entities

Figure 14 depicts the “acknowledged countries” (i.e., the the number of papers per country of affiliation of entities acknowledged), while Figure 16 provides more details on the number of times entities were acknowledged.

J Relation between Tasks and Implicitly Benefitting Stakeholders

Figure 17 presents a heat-map that links the high-level tasks to the combined explicitly and implicitly benefitting stakeholders.

K Risks, Concerns and Limitations Breakdown

Figure 19 distinguishes three levels of engagement with stated risks: High (directly mitigated or discussed in substantial depth), Middling (discussed as part of future work), and Low (briefly mentioned only). Across most risk categories, the majority of engagement is at a Low or Middling level. Methodology limitations show 90% Middling engagement – they are widely acknowledged but rarely addressed in the current work. Dataset limitations are 56%

EU-wide entities acknowledged in 10 papers.

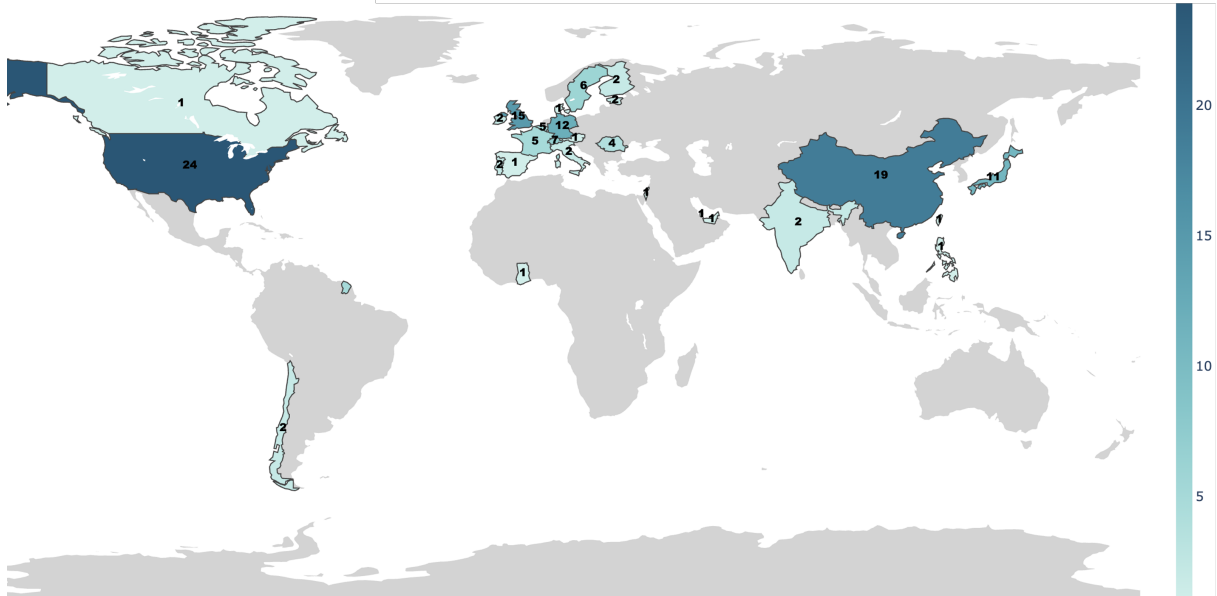


Figure 14: Acknowledged countries (i.e., the number of papers per country of affiliation of entities acknowledged in the surveyed papers).

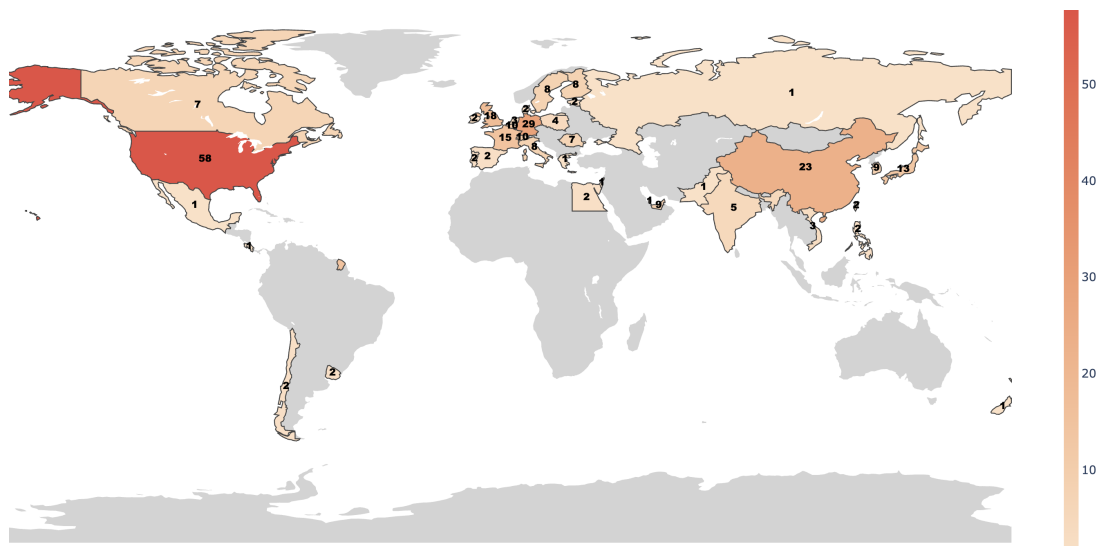


Figure 15: Author countries (i.e., the number of papers per country of author affiliation).

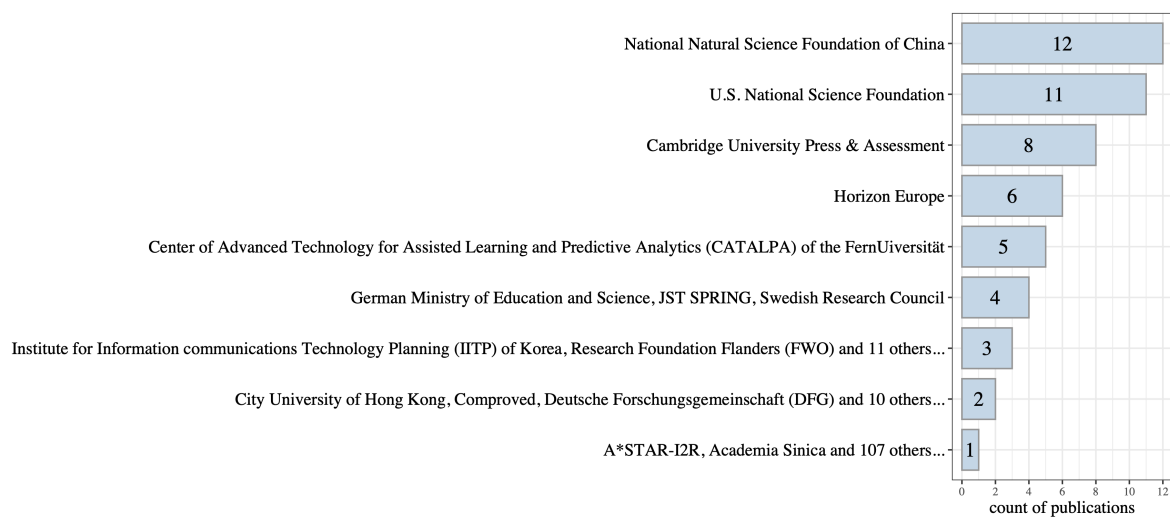


Figure 16: Acknowledged entities (i.e., the number of times an entity was acknowledged in the surveyed papers).



Figure 17: Heat-map relating the high-level tasks to the combined explicitly and implicitly benefitting stakeholders.

Paper ID	Annotator 1	Annotator 2	Annotator 3	Required elements	Computation
Task					
2024.nlp4call-1.14	Conversational intelligent tutoring system (ITS) for EFL speakers	Intelligent tutoring systems	Conversational intelligent tutoring system (ITS) for L2 English	• ITS	ITS mentioned in all three annotations → agreement = 100%
2025.bea-1.2	LLM vs. human proofreading of L2 writing	Proofreading in L2 writing (automated error correction)	Analysis of lexical and syntactic interventions of human and LLM proofreading aimed at improving intelligibility in identical L2 writings	• Human vs. LLM proofreading • L2 writing	Element of “human vs. LLM” not mentioned by Annotator 2 → agreement = 66.67%
Methods					
2024.bea-1.58	Compare zero-shot and few-shot prompting with LLMs vs. fine-tuned models for assessing short answers.	Zero-shot and few-shot prompting (with some fine-tuning) for automated scoring	Use of LLMs (GPT and LLaMA) for automated scoring of short answer responses	• Comparison between zero-shot and few-shot • Use of fine-tuned models	Element of “zero-shot vs. few-shot” not mentioned by Annotator 3 → agreement = 66.67%
Future deployment					
2025.emnlp-main.992	In teacher-in-the-loop real world applications	In high-stakes assessment scoring	AES systems for English	• Teacher in the loop • High-stakes assessments • AES systems	None of the annotators fully overlap → agreement = 0%

Table 7: Examples of majority agreement computation for free-text annotation dimensions. Possible values: 0% (none of the annotations fully overlap), 66.67% (full overlap for two of the three annotations), or 100% (full overlap for all three annotations).

Middling. Risk of bias, one of the most commonly cited concerns, is engaged at a High level in only 17% of papers that raise it; in 45% of cases it is Middling, and in 38% it is Low.

High engagement is most consistently found in a small set of categories: fair compensation for stakeholders (100% High, though only 6 papers raise this at all), and to a lesser extent data protection (65% High). Risk of hallucination, lack of human evaluation, and the gap between research and real-world application are predominantly engaged at Low or Middling levels – noted as future work, but rarely designed around. This pattern suggests a community that is aware of the ethical dimensions of its work but has not yet developed consistent norms for acting on them within the scope of indi-

vidual papers.

L Areas of Future Work

Figure 20 shows the areas of future work explicitly mentioned in the papers, split across five high-level categories (“stakeholder inclusion”, “technical development”, “expand scope”, “engage with issues”, and “none/not applicable”).

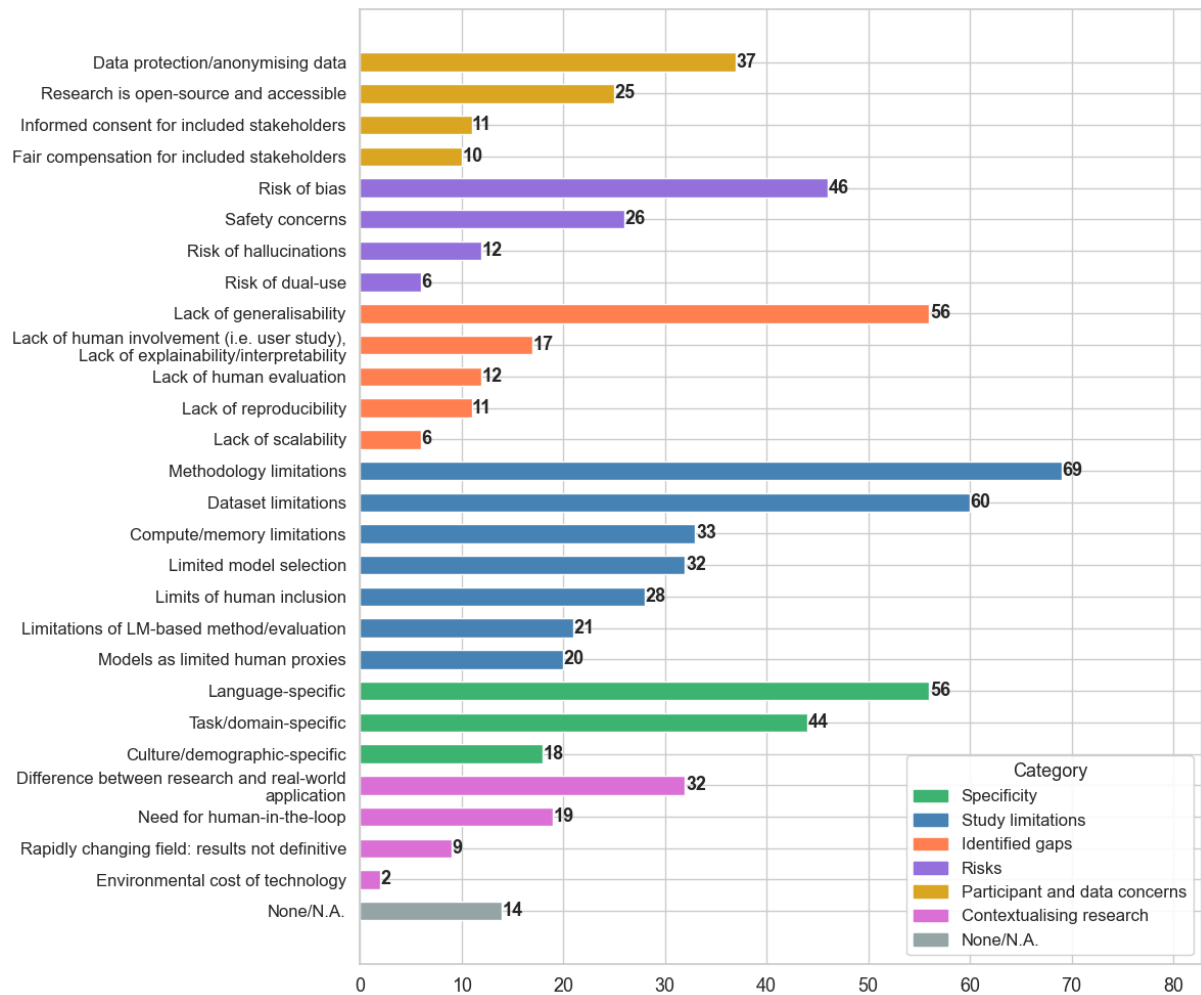


Figure 18: Risks, concerns and limitations explicitly raised by paper authors split across six high-level categories (showing the number of papers; note that a paper may report more than one area of risk).



Figure 19: Engagement levels for the risks, concerns and limitations explicitly raised by paper authors; we distinguish 3 levels of risk engagement: *High* (a risk, concern or limitation that is directly mitigated in the paper or discussed in great depth), *Middling* (discussed as part of future work), and *Low* (briefly mentioned only).

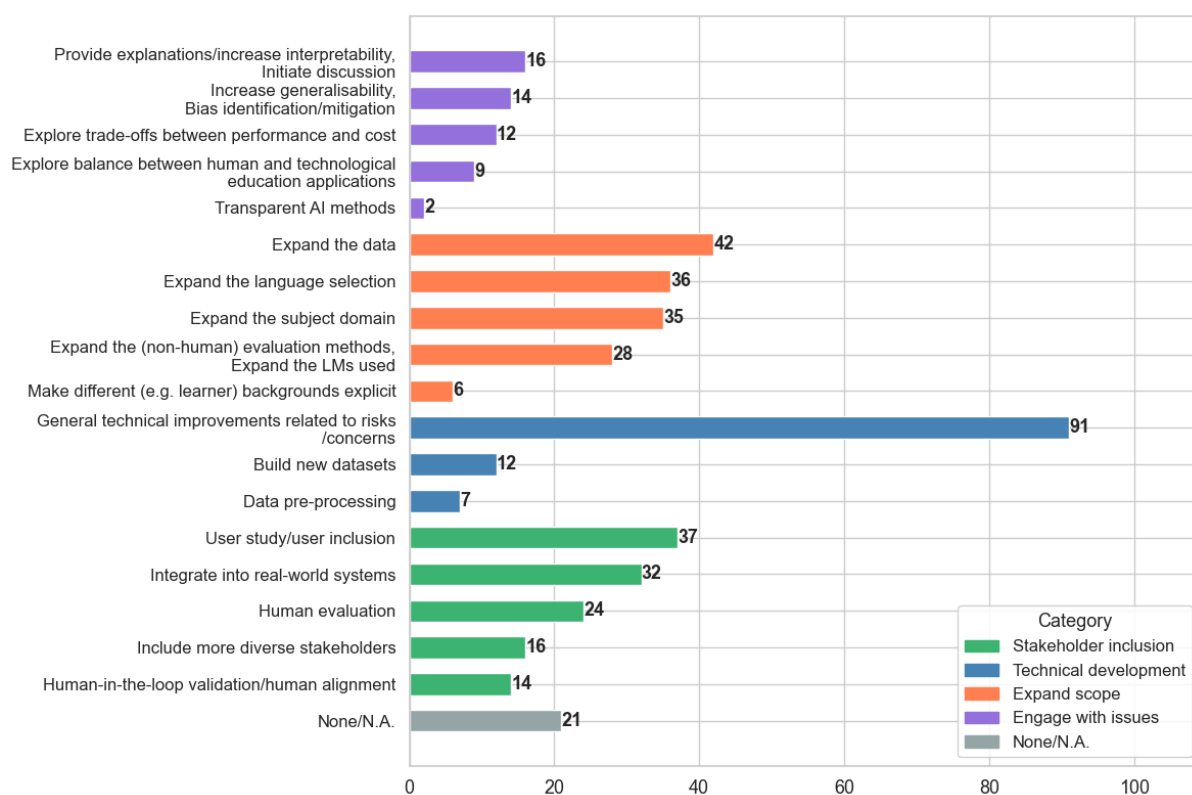


Figure 20: Areas of future work explicitly mentioned in the papers split across four high-level categories (showing the number of papers; note that a paper may report more than one area of future work).

High-level task	Papers
Automated assessment (AES, ASAG)	Yaneva et al. (2024b); Yarmohammadtoosky et al. (2025); Chamieh et al. (2024); De Vrindt et al. (2024); Crossley et al. (2024); Frederick Eneye et al. (2025); Carpenter et al. (2024); Doi et al. (2024); Arronte Alvarez and Xie Fincham (2025); Chiffigarov et al. (2025); Rezayi et al. (2025); Yancey et al. (2024); Bradford et al. (2024); Bannò et al. (2024); Kwako and Ormerod (2024); Löber et al. (2024); Nebhi et al. (2025); Qwaider et al. (2025); Hjortnaes et al. (2024); De Vrindt et al. (2025); Bannò et al. (2025); Stahl et al. (2024); Mirabella and Brunato (2025); Asano et al. (2025); Li et al. (2024); Li and Ng (2024); Wang et al. (2024c); Chen and Li (2024); Chen et al. (2025); Shibata and Miyamura (2025); Srivatsa et al. (2025b); Eltanbouly et al. (2025); Kim et al. (2025a); Lee et al. (2024b); He and Li (2024); Boquio and Naval (2024); Chakravarty et al. (2025); Yoo et al. (2025); Karim et al. (2025); Li and Ng (2025); Su et al. (2025); Chu et al. (2025a); Bexte et al. (2024); Koutchme et al. (2024); Schaller et al. (2024); Muñoz Sánchez et al. (2024a); Bexte and Zesch (2025); Geng and Alfter (2025); Bexte et al. (2025); Bloch et al. (2025); Zesch et al. (2025); Urrutia et al. (2025); Zehner et al. (2025); Elaraby and Litman (2025); Tran et al. (2025); Dascalescu et al. (2025)
Grammatical error correction (GEC)	Staruch et al. (2025); Sorokin and Nasyrova (2025); Vainikko et al. (2025); Luhtaru et al. (2024); Yang and Quan (2024); Wang et al. (2024b); Tang et al. (2024); Goto et al. (2025c); Ye et al. (2025); Goto et al. (2025b); Bhattacharyya and Bhattacharya (2025); Li et al. (2025); Qorib et al. (2024); Koo et al. (2024); Katinskaia and Yangarber (2024); Wu et al. (2024); Alhafni and Habash (2025); Cao et al. (2025); Koyama et al. (2025); Goto et al. (2025a); Wang et al. (2025d); Stahlberg and Kumar (2024); Omelianchuk et al. (2024); Kobayashi et al. (2024); Qiu et al. (2025); Östling et al. (2025); Michael and Horbach (2025); Masciolini et al. (2025); Seminck et al. (2025); Staruch (2025); Tyen et al. (2024); Kucharavy et al. (2025); Galletti and Cesaroni (2025); Koutchme et al. (2025); Marciniak et al. (2025); Petukhova and Kochmar (2025); Tiwari and Rastogi (2025); Correa Busquets et al. (2025)
Text simplification and complexity prediction	Tack (2024); Veeramani et al. (2024); Alfter (2024); Sastre et al. (2024); Padovani et al. (2024); Rooein et al. (2024); Katinskaia et al. (2025); Ribeiro-Flucht et al. (2024); Yousefpoori-Naeim et al. (2024); Wang et al. (2024a); Kelious et al. (2024a); Cristea and Nisioi (2024); Lim and Lee (2024); Yaneva et al. (2024a); Tack et al. (2024); Bulut et al. (2024); Cristea and Nisioi (2024); Shardlow et al. (2024); Enomoto et al. (2024); Dutilleul et al. (2024); Goswami et al. (2024); Strohmaier and Buttery (2024); Übertück-Fries et al. (2024); Kelious et al. (2024b); Alfter (2025); Miyata et al. (2025); Vu et al. (2025); Degraeuwe (2025)
Intelligent tutoring system (ITS)	Pérez-Ortiz et al. (2024); Yasser et al. (2025); An et al. (2025); Kochmar et al. (2025); Park et al. (2025); Wang et al. (2025a); Ribeiro-Flucht et al. (2025); Almasi and Kristensen-McLachlan (2025); Kucheria et al. (2025); Lee et al. (2024a); Pal Chowdhury et al. (2025b); Ikram et al. (2025); Siyan et al. (2024); Wang et al. (2025b)
Content generation	Bodnar (2025); Benedetto et al. (2025); Leite and Lopes Cardoso (2025); Paddags et al. (2024); Berruti et al. (2024); Ma et al. (2025); Liu et al. (2025); Durward and Thomson (2024); Jiang et al. (2025); Wang et al. (2025c); Scaria et al. (2024); Ashok Kumar and Lan (2024); Scarlatos et al. (2024); Stowe et al. (2024); Glandorf and Meurers (2024); Nikolova-Stoupak et al. (2024); Toussaint et al. (2024); Säuberli et al. (2025); Parikh et al. (2025); Kim et al. (2025b); Huovinen and Hämäläinen (2025); Poon et al. (2025)
Datasets, data mining and knowledge extraction	Sung et al. (2025); Akef et al. (2025); Chitez et al. (2025); De Kuthy et al. (2025); Han and Choi (2025); Akter et al. (2025); Sharma and Zhang (2025); Chen and Zhao (2025); Foret et al. (2024); Yang et al. (2025); Velentzas et al. (2024); Rozovskaya (2024); Zhao et al. (2025); Mita et al. (2024); Ding et al. (2024); Beigman Klebanov et al. (2024); Schaller et al. (2025); Dumitran et al. (2025); Mulcaire and Madnani (2025)
Knowledge tracing and learner modelling	Gurin Schleifer et al. (2024); Cristea and Nisioi (2024); Martynova et al. (2025); Srivatsa et al. (2025a); Stearns et al. (2024); Chu et al. (2025b); Hayat et al. (2024); Shi and Mangalam (2025)
Other	Gupta et al. (2025); Ilagan et al. (2024); Schmalz and Tack (2025); de Chillaz et al. (2025); Singh et al. (2025); Muñoz Sánchez et al. (2024b); Ballier and Méli (2024); Degraeuwe and Goethals (2024); Ayari and Li (2024); Hülsing and Horbach (2024); Pal Chowdhury et al. (2025a); Skidmore et al. (2025); N J et al. (2025); Kolagar et al. (2025); Shimabukuro et al. (2025); Sakunkoo and Sakunkoo (2025); Mao et al. (2025); Uluslu and Schneider (2025)

Table 8: Table showing the mapping from high-level tasks to individual papers and their specific tasks.

Children’s English Reading Story Generation via Supervised Fine-Tuning of Compact LLMs with Controllable Difficulty and Safety

Qian Shen

University of Florida
qian.shen1@ufl.edu

Fanghua Cao*

University of Florida
cao.fanghua@ufl.edu

Min Yao*

University of Florida
min.yao@ufl.edu

Shlok Gilda

University of Florida
shlogilda@ufl.edu

Bonnie J. Dorr

University of Florida
bonniejdorr@ufl.edu

Walter L. Leite

University of Florida
leitewl@ufl.edu

*Equal contribution

Abstract

Large Language Models (LLMs) are widely applied in educational practices, such as for generating children’s stories. However, the generated stories are often too difficult for children to read, and the operational cost of LLMs hinders their widespread adoption in educational settings. We used an existing expert-designed children’s reading curriculum and its corresponding generated stories from GPT-4o and Llama 3.3 70B to design different experiments for fine-tuning three 8B-parameter LLMs, which then generated new English reading stories that were subjected to quantitative and qualitative evaluation. Our method prioritizes controllability over scale, enabling educators to target reading levels and error patterns with a compact, affordable model. Our evaluation results show that with appropriate fine-tuning designs, children’s English reading stories generated by 8B LLMs perform better on difficulty-related metrics than those from zero-shot GPT-4o and Llama 3.3 70B, with almost no discernible safety issues. Such fine-tuned LLMs could be more broadly used by teachers, parents, and children in classrooms and at home to generate engaging English reading stories with children’s interests, controllable difficulty and safety.

1 Introduction

Children’s reading fluency is a key marker of reading development and profoundly influences language comprehension, expressive ability, and academic achievement (Spear-Swerling, 2006; Veenendaal et al., 2015). For young learners, varied forms of repeated reading have been shown to effectively boost oral reading fluency (Rasinski et al., 2016). Consequently, empirical work indicates that providing diverse, engaging materials elevates children’s interest and motivates more practice (Merisuo-Storm and Soininen, 2014). Yet it is clearly difficult for teachers and researchers to manually assemble large volumes of content tai-

lored to each child’s preferences. Generative AI offers a promising way to meet this need.

Today, generative AI can be trained to craft engaging, age-appropriate narratives that reflect diverse traditions and to independently produce imaginative text suited for advanced image-generation systems, enriching the storytelling experience (Lesner et al., 2024). Such stories can resonate with children’s imaginative worlds while offering educational and entertaining narratives aligned with the moral values appropriate to different age groups (Leite et al., 2025). Creating stories tailored to children’s interests and needs may enhance motivation, increase reading engagement, and help achieve the goal of improving reading skills (Al-Siyabi, 2024).

Prior work (Leite et al., 2025) used zero-shot prompting based on expert-authored curricula to generate children’s stories using OpenAI’s GPT-4o and Meta’s Llama-3.3 70B (Leite et al., 2025). Evaluated against pedagogical theories and NLP metrics, the results revealed a critical limitation: even state-of-the-art closed-source models like GPT-4o struggle with strict pedagogical constraint satisfaction. Specifically, they frequently fail to reliably restrict lexical and syntactic difficulty to target lower grade levels (e.g., K-2 readability scores) without degrading narrative coherence. Furthermore, the recurring API costs of closed models and the massive hardware requirements to deploy 70B+ models locally present prohibitive barriers to widespread educational adoption (Irugalbandara et al., 2024).

To democratize access to personalized educational content, deploying compact, local LLMs (under 10B parameters) is an intuitive solution. However, adapting sub-10B models for this task is not merely a standard Supervised Fine-Tuning (SFT) problem; it presents a unique scientific challenge. Compact models typically suffer from a severe trade-off between narrative creativity and strict rule-following. When conditioned on multi-

dimensional pedagogical constraints (e.g., strict vocabulary boundaries, syntax simplification, and safety guardrails), small models are prone to mode collapse or logical fragmentation. Standard SFT alone is insufficient to bridge this "controllability gap."

To address this gap, we present a systematic empirical comparison of various fine-tuning and inference strategies for phoneme-controlled story generation. By utilizing high-quality stories, we evaluate how effectively sub-10B models can internalize strict pedagogical alignment through techniques such as standard SFT, reward-weighted SFT, and augmented input. Our primary objective is to provide empirical evidence that, with appropriate training pipelines, compact models can achieve controllable difficulty and safety that match or even surpass those of massive zero-shot 70B models. This empirical foundation demonstrates the viability of broader, equitable adoption of AI-driven reading interventions irrespective of economic or hardware constraints.

Accordingly, our research is guided by two questions: (RQ1) Which SFT strategies most effectively enable sub-10B LLMs to generate children's stories with controllable reading difficulty that meets K–2 pedagogical standards? (RQ2) Can these compact fine-tuned models achieve levels of content safety comparable to those of state-of-the-art zero-shot 70B models?

2 Related Work

We surveyed the current research works on the applications of generative AI, controllable story generation, and supervised fine-tuning in educational practice to identify available methods and pinpoint the research gap.

2.1 AI-Generated Stories for Children

As generative AI continues to be applied in various fields, researchers have explored using multimodal generative AI, spanning text, image, and voice generation, to construct children's story generation systems (Chowdhury et al., 2025; Liu et al., 2025). For example, Han and Cai (2023) propose AISTory, a visual narrative prototype for children's creative expression and literacy, while Fan et al. (2025) develop StoryPrompt to enable child–AI co-creation of stories and comics.

Despite these advances, studies reveal persistent limitations. Sun et al. (2024) find that AI-based

storytelling still falls short of parents' expectations due to interaction and algorithmic challenges. Jin and Yuan (2025) highlight that consistency between stories and illustrations, accuracy relative to the real world, and clear emotional expression are crucial for educational use, while content safety remains a fundamental requirement. Liao (2025) further points out that AI-generated stories often contain language too complex for young children. In summary, as LLMs are increasingly deployed for children's story generation, age-appropriate readability, and content safety have become critical concerns that the field has yet to sufficiently address.

2.2 Automatic Story Generation with Controllable Difficulty

Recent advances in LLMs and SFT technologies have enabled clearer control over text readability (Glandorf and Meurers, 2024; Kim et al., 2025), yet a critical challenge remains in educational contexts: the tension between strict early-elementary readability constraints and narrative diversity. Rigid constraints such as those required for phonics curricula often force language models into repetitive, formulaic patterns (Holtzman et al., 2020), a problem particularly pronounced in compact local models with limited capacity to balance linguistic creativity with vocabulary restrictions (Farajidizaji et al., 2024). Addressing this trade-off requires embedding pedagogical constraints directly into model parameters, motivating the use of SFT in our work.

2.3 Application of Supervised Fine-Tuning in Educational Practice

SFT has been widely adopted to adapt LLMs to concrete educational tasks. At the assessment level, Duong-Trung et al. (2024) propose BloomLLM, which fine-tunes an LLM on question–answer pairs labeled with Bloom's taxonomy, enabling controllable generation across cognitive levels. In student support, Assayed et al. (2024) fine-tune a BERT-based model on domain-specific advising data, improving contextual accuracy over untuned baselines. More broadly, Chen et al. (2023) report that fine-tuned LLMs outperform prompt-only models on tasks such as explanation generation and content recommendation. SFT has also been integrated into interactive learning systems: Gao et al. (2025) fine-tune an LLM within Tailor-Mind to support self-regulated learning, with user studies showing improved goal setting and monitoring. Building on

the general paradigm of aligning pre-trained models to human intent through SFT (Ouyang et al., 2022), these works collectively suggest that SFT offers a practical route to embed pedagogical structure, domain specificity, and safety constraints directly into model parameters.

3 Methodology

Through various data preprocessing methods, we constructed multiple SFT experiments on three 8B LLMs using 2,580 stories generated from GPT-4o and Llama-3.3-70B, and generated new English reading stories for children with fine-tuned LLMs. Then, we selected several evaluation metrics to analyze the controllability of story difficulty and safety, and other metrics such as the similarity between stories generated in each experiment. Figure 1 and the following sections detail our experiments, reasoning, and evaluations.

3.1 Data

Our data derive from a K–2 English reading curriculum from the University of Florida Literacy Institute (UFLI) (Lane and Contesse, 2022): 129 progressively difficult lessons, each specifying target phonemes. Figure 4 in Appendix B shows an example of this curriculum. With zero-shot prompting, GPT-4o and Llama-3.3 70B were used to generate 10 stories per lesson constrained to the designated phonemes, yielding 2,580 stories (Leite et al., 2025). Although these models struggle with strict pedagogical constraint satisfaction in zero-shot settings, their outputs still provide sufficient linguistic quality and narrative coherence to serve as training targets; the pedagogical alignment is then refined through the SFT process. Each story contains 50–350 words, with word counts increasing with lesson difficulty.

3.2 Supervised Fine-Tuning Experiments

We established a baseline method using direct SFT on LLMs with our data, and designed three SFT experiments based on different data preprocessing methods.

3.2.1 Model and Parameter Setting

In our SFT experiments, we fine-tuned three LLMs: Llama 3 8B, Apertus 8B-instruct-2509, and Granite 3.3 8b-instruct. To adapt the base LLM efficiently, we used QLoRA (Dettmers et al., 2023) (4-bit NF4 quantization + LoRA adapters) for parameter-efficient fine-tuning. The LLMs were quantized

into 4-bit NormalFloat (NF4) format with double quantization and trained using LoRA adapters (rank = 32, α = 64, dropout = 0.1, learning rate = 1e-4, batch size = 4, epochs = 5, warmup ratio = 0.03, and the optimizer is AdamW) attached to attention and MLP projection layers. Training is performed in BF16 precision with gradient checkpointing to reduce the memory footprint.

3.2.2 SFT Experimental Design

- **Baseline:** We adopt the simple SFT experiment as our baseline. Specifically, we construct input–target pairs by using the phonemes contained in the corresponding curriculum lessons as inputs and the 2,580 previously generated stories as targets, and fine-tune three selected LLMs separately with QLoRA. We then use the fine-tuned LLMs to generate 10 stories for each lesson.
- **"Good Stories":** Some research works on fine-tuning LLMs believe that using higher-quality data can outperform using larger amounts of lower-quality data (Chen et al., 2024; Zhou et al., 2023). Accordingly, we first computed five metrics for the 2,580 stories previously generated by GPT-4o and Llama 3.3 70B (we detail their computation later): Spache readability, language model perplexity (LM-PPL), coherence, syntactic complexity, and toxicity. To avoid an overly small training set, we selected stories that were not worse than the corpus means on all metrics, yielding 996 stories in total, and used them to fine-tune the three LLMs, constructing input–target pairs from the corresponding phonemes and stories for SFT.
- **Rewarded SFT:** We considered Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022) as a fine-tuning approach, but our sample size was insufficient to train an effective reinforcement learning (RL) model. Thus, we follow LLM alignment approaches that reinterpret RL as supervised learning with reward-dependent weights (Peters, 2006; Mukherjee et al., 2025). Concretely, each training example is assigned a scalar reward based on the five metrics we mentioned, and the standard cross-entropy loss in the SFT Trainer would be reweighted accordingly. It is closely related to reward-weighted regression schemes explored for offline LLM alignment

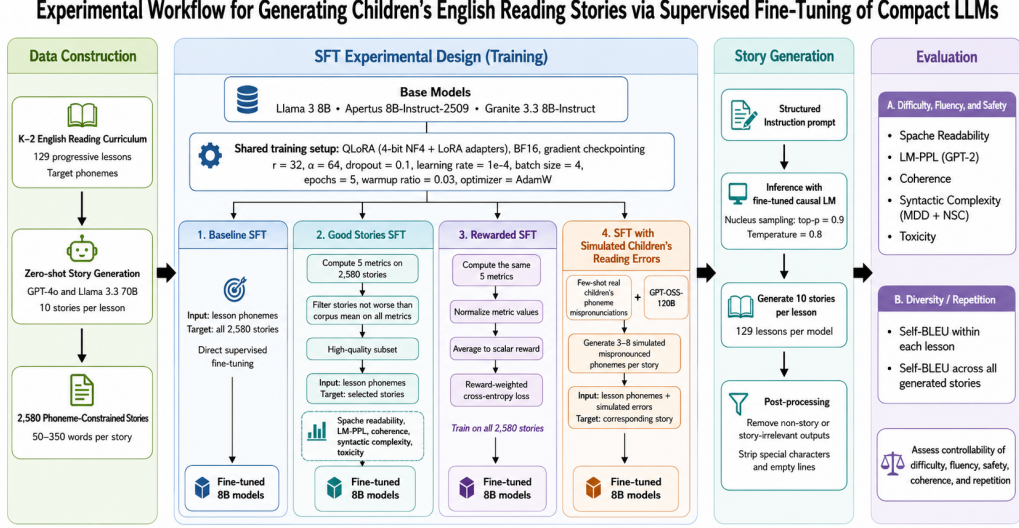


Figure 1: System architecture and experimental workflow for generating children’s English reading stories via supervised fine-tuning of compact LLMs.

and self-improvement (Qu et al., 2024), and fits within the broader “learning from rewards” framework for LLMs (Zhang et al., 2025). As in our baseline, we construct input–target pairs by using the phonemes and all the 2,580 stories. During training, we compute a scalar reward r_i by combining five automatic evaluation metrics. Since these metrics are defined on different scales and directions, each metric is first normalized into $[0, 1]$. For metrics where lower values indicate better quality (e.g., readability difficulty or perplexity), we apply inverted min–max normalization:

$$\tilde{m}_i = \max\left(0, \min\left(1, \frac{b_m - m_i}{b_m}\right)\right),$$

where m_i is the raw metric value and b_m is a predefined upper bound. For metrics where higher values indicate better quality, we apply standard min–max normalization. The final scalar reward is computed as the unweighted average of the normalized metrics:

$$r_i = \frac{1}{5} \sum_{k=1}^5 \tilde{m}_i^{(k)}.$$

This formulation ensures comparability across metrics while preserving stability during training.

- SFT with simulated children’s reading errors: Inspired by the Self-Instruct paradigm

(Wang et al., 2023), we leverage advanced LLMs to simulate diverse phonemic errors typical of early readers, addressing the data scarcity of real-world error samples. Following the LaMP framework for personalization (Salemi et al., 2024), we constructed a few-shot learning setup using data on real children’s phoneme mispronunciations observed while reading several stories, and used OpenAI’s GPT-OSS-120B (Agarwal et al., 2025) to generate 3–8 simulated child mispronounced phonemes for each of the 2,580 stories. Then, we used the phonemes from each lesson and simulated child mispronounced phonemes as input, and the 2,580 stories as the target to do SFT. This approach aligns with recent findings from “STaR: Bootstrapping Reasoning with Reasoning” (Zelikman et al., 2022), which suggest that training on high-quality, educationally targeted synthetic data significantly enhances model performance in specialized domains.

3.3 Evaluation Metrics

To evaluate whether the difficulty and safety of stories generated by fine-tuned LLMs are controllable enough, we employed five evaluation metrics to measure these stories’ difficulty and safety for children. First, we used the Spache Readability score (Spache, 1953) to assess text difficulty specifically for our target audience of early elementary readers (up to the 3rd grade), where a score below

3.0 shows text at approximately the second-grade reading level.

Second, to measure the fluency and naturalness of the generated stories, we computed LM-PPL. We employed the GPT-2 model via the Hugging Face Evaluate library to calculate the perplexity.

Third, we introduced a coherence score to quantify local narrative consistency by averaging the number of shared named entities between consecutive sentences, identified using spaCy. Fourth, we evaluated syntactic complexity to capture the structural difficulty of the text. We designed a composite metric that combines the average maximum dependency distance (MDD) within sentences and the average number of subordinate clauses (NSC), computed using spaCy.

Also, text toxicity was assessed to ensure the safety of content intended for children. We employed Detoxify (Hanu and Unitary team, 2020) to identify potential offensive or harmful language, where a score closer to 0 indicates safer content, aiming to filter out stories that exhibit high probabilities of toxic content.

Because the stories generated by GPT-4o and Llama 3.3 exhibit some content-level repetition, we also measured the degree of inter-story repetition among stories produced by the fine-tuned LLMs. We used Self-BLEU (Zhu et al., 2018) to measure the degree of repetition in the generated stories. It is able to evaluate each story as a hypothesis while treating the remaining stories as references. Thus, higher scores show stronger similarity across outputs and lower generative diversity. To capture repetition at different levels of granularity, we compute Self-BLEU both within each lesson and across the generated stories of each LLM in each experiment.

For a given story belonging to a lesson, we can compute Self-BLEU by using all other stories in the same lesson as references. It reflects how likely the model is to reuse similar expressions, narrative structures, or lexical patterns when conditioned on similar lesson inputs.

In addition to this local view, we measured the repetition between 1,290 stories of each LLM in each experiment by treating every other story as a reference for each story. This provides a broader perspective on whether the model collapses to a narrow set of global templates across lessons. Formal mathematical definitions of all evaluation metrics are provided in Appendix A.

4 Results

In our generation stage, we perform inference-only decoding with a fine-tuned causal language model. We apply the LoRA-adapted model parameters to the three base LLMs and generate text autoregressively, conditioned on a structured instruction prompt. Text generation uses nucleus sampling (top-p=0.9) with temperature (0.8), which produces stories without modifying model parameters. Since some outputs from the fine-tuned LLMs contained content extraneous to the stories, and in some cases, the entire output was not a story, we identified these samples through manual inspection, flagging outputs that lacked a coherent narrative structure, consisted mainly of prompt repetition or metadata, or were otherwise not readable as stories. Additionally, to avoid affecting the computation of our story evaluation metrics, we stripped special characters and meaningless blank lines or whitespace from the LLMs' outputs.

4.1 Quantitative Evaluation

Table 1 shows the evaluation results for stories generated by LLMs fine-tuned under Rewarded SFT and Baseline. Examining these results, we find that, except for the overall repetition rate of stories from the Rewarded SFT experiment and the LM-PPL of stories from the Good Stories experiment and SFT with simulated children's reading errors, the metric scores for stories from these three experiments show improvements relative to the baseline.

Apart from overall repetition, stories from Rewarded SFT and SFT with simulated children's reading errors often approach or surpass those from GPT-4o and Llama 3.3 70B on other evaluation metrics. Across the three 8B LLMs we examined (except for our baseline experiment), stories generated by Llama 3 always underperform those from the other two LLMs on most evaluation metrics. In contrast, the evaluation results of stories produced by Granite 3.3 and Apertus do not show marked differences between the various experimental designs.

The Spache readability scores of stories generated by our fine-tuned LLMs are mostly lower than those from GPT-4o and are comparable to or even lower than those from Llama 3.3. However, this average Spache readability score still has significant room for improvement to accommodate the reading abilities of children below the second-grade level. The mean toxicity of all our generated stories is

very close to 0, suggesting that only a minimal number of stories might pose a safety risk.

In our experiments, the stories generated by LLMs fine-tuned with Rewarded SFT exhibited high stability, being almost entirely free of hallucinations, garbled text, story-irrelevant outputs, or null values. In the other experiments, while the stories from fine-tuned LLMs did contain a small portion of these undesirable outputs, the proportion was less than 10% of the 1,290 stories generated per model in each experiment.

To further evaluate whether the evaluation metric values for stories generated by LLMs fine-tuned in our SFT experiments differ significantly, in a statistical sense, from those of stories produced by GPT-4o and Llama 3.3 70B, we conducted significance testing. Because the stories are non-identical and our experiments cannot fully eliminate LLM hallucinations or errant responses, although all the stories generated by all the experiments being compared are usable, we used Welch’s T-test to ensure the robustness of the test results.

Given that the toxicity of the story is nearly zero across the board and that the two metrics related to repetition are not consistently standardized across datasets, we compared the corresponding metric values of fine-tuned LLM stories that achieve the best mean performance on coherence, syntactic complexity, Spache readability, and LM-PPL against those of GPT-4o and Llama 3.3 70B. In the results of Welch’s T-test, all p-values are less than 0.001, and all Cohen’s d are greater than 0.8, which shows that the differences in the mean scores for coherence, syntactic complexity, Spache readability and LM-PPL between the stories of fine-tuned LLMs and those of GPT-4o and Llama 3.3 70B are statistically significant.

In summary, quantitative evaluation shows that our SFT experiments, particularly Rewarded SFT, enable fine-tuned 8B LLMs to generate stories with difficulty levels more suitable for K-2 children than those from GPT-4o and Llama 3.3 70B, while presenting almost no discernible safety issues. Complete evaluation results across all four experiments, along with a visual comparison of metric trends across all models and SFT strategies, are provided in Appendix B (Table 2 and Figure 3).

4.2 Qualitative Analysis

In our evaluation of the newly generated stories, we observed outliers in some evaluation metrics. Primarily, the mean LM-PPL values for stories from

SFT with simulated children’s reading errors were higher than those of other experiments. Observation of these stories revealed that their higher LM-PPL values were either due to their brevity, which led to increased uncertainty and higher cross-entropy per word, or because the LLMs appended content irrelevant to the narrative after generating a short story. This caused an abrupt genre shift, thereby increasing the text’s perplexity.

Simultaneously, we observed that some stories in the newly generated sets exhibited toxicity scores significantly higher than others, and we examined these stories closely. We identified several potential factors contributing to the higher toxicity scores, such as the inclusion of language like "you’re too fat," "this smells too stinky," and "the pig can gag," or the appearance of characters portrayed as unfriendly or potentially harmful, such as rats. These elements pose a risk of causing discomfort to children or exposing them to content such as body shaming. Overall, the number of stories containing such content was five or fewer in each dataset of 1,290 stories.

Additionally, the instances of excessively high Spache readability grade scores or low coherence scores in some stories were attributed to the presence of difficult words and numerous line breaks, respectively. The syntactic complexity of some stories was collectively elevated by higher rates of subordination, coordination density, complex nominal phrases, premodification, and parentheticals, as well as unconventional word order and fragmented sentence structures.

Figure 2 in Appendix B shows a "bad" story and a "good" story from our generated stories as an example. The comparison between the two stories reveals notable differences in linguistic suitability for early children. The "good" story maintains clear coherence, following a straightforward sequence of actions that aligns well with the narrative-processing abilities of young children. Events progress linearly and predictably, and character references remain stable, enabling developing readers to follow the plot without excessive inferential demands. In contrast, the "bad" story introduces a broad and descriptively dense setup, then shifts abruptly toward unrelated or ill-formed fragments. These discontinuities reduce overall coherence and make the narrative less suitable for young readers who benefit from a tighter event structure.

In terms of syntactic complexity, the "good" story employs short, regular sentence structures

Table 1: Evaluation results of stories generated by LLMs. Values are reported as Mean (SD). $\uparrow$ indicates higher is better; $\downarrow$ indicates lower is better.

Metric	Baseline			Rewarded SFT		
	Llama 3	Granite 3.3	Apertus	Llama 3	Granite 3.3	Apertus
Coherence $\uparrow$	0.02 (0.05)	0.07 (0.05)	0.09 (0.13)	0.12 (0.14)	0.18 (0.18)	0.13 (0.15)
Syntactic Complexity $\downarrow$	4.63 (2.02)	3.72 (1.05)	3.41 (0.90)	3.38 (1.11)	3.12 (0.88)	2.96 (0.98)
Spache Readability $\downarrow$	4.05 (1.04)	3.52 (0.93)	2.83 (0.47)	2.71 (0.70)	2.56 (0.60)	2.34 (0.68)
Toxicity $\downarrow$	0.01 (0.04)	0.06 (0.12)	0.00 (0.01)	0.01 (0.02)	0.02 (0.06)	0.02 (0.07)
LM-PPL $\downarrow$	23.16 (13.22)	24.91 (10.21)	16.49 (3.88)	16.86 (4.29)	14.55 (3.56)	19.73 (6.64)
Repetition in lessons $\downarrow$	0.03 (0.03)	0.11 (0.09)	0.12 (0.06)	0.11 (0.06)	0.12 (0.07)	0.09 (0.08)
Total repetition $\downarrow$	0.21 (0.09)	0.37 (0.16)	0.40 (0.08)	0.37 (0.10)	0.42 (0.11)	0.32 (0.13)

characteristic of early-grade leveled texts. The “bad” story, by comparison, alternates between long, multi-clause sentences and incomplete constructions, creating variability that may be less aligned with the decoding patterns expected at this developmental stage. The lexical difficulty further highlights this contrast: the “good” story uses high-frequency, concrete vocabulary appropriate for K–2 learners, whereas the “bad” story incorporates abstract, low-frequency terms (“spectacle,” “insatiable,” “peculiar”) that can make comprehension more challenging for beginning readers. These lexical choices also contribute to higher expected perplexity, indicating less predictable and less contextually grounded language.

Beyond outliers in individual metrics, we also examined the pattern of inter-story repetition reflected in Self-BLEU scores. As shown in Table 1, for most generated story datasets (every 1,290 stories per LLM in each experiment), these scores indicate suboptimal diversity. Upon inspection of the generated stories, we observed the recurrence of certain names, animal names, locations, and activities across multiple narratives. Even combinations of these recurring entities, such as Sam, Pam, mats, pigs, and farm, also reappear. Furthermore, some stories feature the same place name repeatedly, albeit attributed to different states within the respective narratives.

Overall, while both stories are grammatically interpretable, the “good” story is more closely aligned with early-elementary reading expectations, whereas the “bad” story exhibits features across coherence, syntax, and vocabulary that render it comparatively less suitable for this age range.

5 Discussion

The repetition patterns observed above are likely attributable to the characteristics of the source train-

ing data. We believe that these phenomena are related to the source stories from GPT-4o and Llama 3.3 70B themselves. In the stories from Llama 3.3 70B, we also observed similar patterns, such as the recurrence of content involving Sam, Pam, mats, pigs, and farm. This issue arose because we could only use a small number of keywords for batch-generating stories with Llama 3.3 70B due to the limitations of the insufficient data collected at the time. The stories from GPT-4o, which used varied keywords, exhibited lower content repetition; however, they also featured recurring place names, consistently attributed to different states such as Ohio, Illinois, or Kansas. This indicates that after fine-tuning, the generative preferences of these 8B LLMs are strongly influenced by the stories used for the fine-tuning process. It aligns with the findings of Santurkar et al. (2023) and Chung et al. (2024), who demonstrate that the specific task patterns and behavioral attributes encoded in fine-tuning data directly dictate the model’s downstream generation capabilities.

In our evaluation process, while we have utilized quantitative metrics based on NLP theory to assess the stories, feedback, and opinions from experts, teachers, parents, and children are still lacking. Incorporating expert ratings, focus group interviews, and collecting feedback from users of the story-generating LLMs would enable us to better, from the perspectives of data and prompt engineering, guide the LLMs to learn the characteristics of stories that are more suitable for children’s needs and to generate new stories more effectively.

In practical classroom applications, stories generated by the fine-tuned 8B LLMs we used are more suitable in difficulty for children from kindergarten to 2nd grade compared to those from the GPT-4o API and the Llama 3.3 70B model, and they present no apparent safety issues. Compared to APIs that require payment and 70B LLMs that

demand substantial hardware resources for deployment, both the fine-tuning and inference costs for 8B LLMs are significantly lower, allowing them to be designed as back-end models for websites and software. Teachers, parents, and children can use these fine-tuned, local models for free in the classroom or at home to generate stories suitable for children’s reading.

However, 8B LLMs are still too large for deployment on mobile devices or for users who wish to fine-tune the models with their own data for unique pedagogical tasks. Additionally, the previously identified issues within the generated stories, such as some inappropriate content and complex syntax, need to be resolved. Furthermore, our current experimental designs are mutually independent, and the effects of combining these designs remain an area for further exploration.

To ensure the stories generated by these LLMs better meet the needs of teaching and learning, as we have previously mentioned, we still require more stories generated from keywords by teachers, parents, and children, authentic data on children’s reading errors, expert ratings of the stories, and any form of user feedback.

6 Conclusion

LLMs have been widely utilized in educational practices such as children’s story generation. We have used GPT-4o and Llama 3.3 70B to generate English reading stories for children from kindergarten to second grade, based on an expert-authored curriculum. However, the cost and hardware requirements of LLMs hinder their widespread implementation in every classroom. We designed three SFT experiments to fine-tune three popular 8B LLMs and generated new stories based on the same curriculum. We evaluated these stories from both quantitative and qualitative perspectives, and discussed the existing problems and the challenges of applying these fine-tuned LLMs in educational practice.

Our constructed SFT experiments are more effective at generating children’s English reading stories with controlled difficulty and safety than the direct SFT baseline. Specifically, stories generated by the 8B LLMs after being fine-tuned in our experiments surpass those from zero-shot GPT-4o and Llama 3.3 70B on difficulty-related evaluation metrics, and they present almost no apparent safety issues. Our "Rewarded SFT" experiment performed com-

paratively better, generating highly stable content that was nearly devoid of any story-irrelevant material. Among the three 8B LLMs we selected, Llama 3’s performance was relatively weaker than the other two.

Our fine-tuned, compact LLMs will further empower more classrooms and families to freely and conveniently generate English reading stories for children from kindergarten to second grade, ensuring these stories possess controllable reading difficulty and are appropriate in their safety for this age group.

In future research work, we plan to collect more stories generated by people using these LLMs with keywords of interest and combine this with expert ratings and feedback from users of the LLMs to construct a complete RLHF pipeline. This will enable us to use higher-quality data, more human feedback, and more effective LLMs fine-tuning strategies to better fine-tune LLMs for generating stories more suitable for children. Furthermore, we will also consider achieving good story generation performance with smaller LLMs through methods such as model distillation, enabling our fine-tuned LLMs to be used by more teachers, parents, and children in classrooms, at home, on computers, mobile phones, and tablets, regardless of their economic situation or hardware resources.

Limitations

In addition to the points already discussed, our research also has some other limitations. First, due to computational resource constraints, we were limited to three L4 GPUs, which prevented us from employing certain SFT strategies in our experimental design; hyperparameters (e.g., the number of training epochs) were also constrained. Second, because related projects were progressing concurrently, we lacked sufficient sample size prior to initiating this study to apply certain fine-tuning techniques, such as completing a full RLHF pipeline. Third, due to IRB review and constraints on practical deployment, our study was built entirely on batch-generated stories and lacks authentic narratives from children, parents, and teachers. Similarly, the generated stories lack expert human evaluation to corroborate the NLP-based automatic evaluation. Moreover, because we used only a single expert-designed curriculum for children’s English reading education, further work is needed to extend our approach to additional curricula and more

languages. Furthermore, automated phoneme coverage evaluation remains an open challenge, as standard grapheme-to-phoneme tools struggle with out-of-vocabulary items and irregular orthographic patterns commonly produced by LLMs. We have attempted g2p-en and NLTK but the non-standard curriculum notation (e.g., "CVC words") and large phoneme sets per lesson produced near-zero hit rates.

Furthermore, in our Rewarded SFT experiment, the scalar rewards were computed using the same automatic metrics later used for evaluation, which raises the possibility that the model was optimized toward the evaluation proxies rather than toward true educational quality. We consider this a limitation and note that human evaluation would be necessary to rule out such over-optimization. Additionally, the comparisons between fine-tuned 8B LLMs and GPT-4o and Llama 3.3 70B are not fully controlled, as the larger models were evaluated only under zero-shot prompting conditions without prompt optimization or few-shot examples. The observed performance differences should therefore be interpreted in this context rather than as a general claim about model capability. Additionally, the fine-tuned models were evaluated on stories generated from the same 129 lessons used during training, rather than on a held-out set of unseen phoneme constraints. This limits our ability to assess generalization to novel lesson inputs.

Ethics

This work involves the generation of educational content for children and therefore requires particular care regarding data provenance, child safety, and deployment. Our training data are derived from an expert-designed K–2 reading curriculum and from stories synthetically generated by large language models, rather than from children’s personal data or identifiable records. In addition, our use of a small, authentic sample of children’s reading errors has been approved by the Institutional Review Board of University of Florida (Protocol # IRB202401327). The simulated reading-error inputs used in one experimental condition are synthetic approximations intended to model common early-reader phonemic mistakes, not diagnostic judgments about any individual child. Nevertheless, synthetic data may encode biases, repetitive patterns, or inappropriate language inherited from source models, and our own analysis found a small

number of outputs containing potentially harmful expressions, including insults, disgust-related language, and body-shaming content. For this reason, these models should not be deployed as fully autonomous systems for children. Responsible deployment should include human oversight by teachers, parents, or other adult supervisors; automatic safety filtering for toxicity and related harms; conservative blocking or review policies for outputs that include appearance-based insults, humiliation, or other potentially distressing language; and clear disclosure that generated stories may still contain errors or unsuitable content. These systems are best used as assistive tools for draft generation and adaptation, not as unsupervised replacements for educators or caregivers. Future work should strengthen this pipeline through expert human review, broader safety evaluation beyond general toxicity, and governance practices that ensure children’s well-being remains the primary criterion for deployment. In preparing this manuscript, the authors made use of AI-assisted writing tools for language polishing. All scientific content, analyses, and conclusions are the sole responsibility of the authors.

References

- Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.
- Salama Saif Hamed Al-Siyabi. 2024. Using chatgpt to enrich children literature and enhance their vocabulary repertoire. *International Journal of Research and Studies Publishing*, 5(58):278–296.
- Suha K. Assayed, Manar Alkhatib, and Khaled Shaalan. 2024. [A transformer-based generative ai model in education: Fine-tuning bert for domain-specific in student advising](#). In Azza Basiouni and Claude Frason, editors, *Breaking Barriers with Generative Intelligence. Using GI to Improve Human Education and Well-Being*, volume 2162 of *Communications in Computer and Information Science*, pages 165–174. Springer.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srivasan, Tianyi Zhou, Heng Huang, et al. 2024. Alpaga-sus: Training a better alpaca with fewer data. In

- The Twelfth International Conference on Learning Representations.*
- Yong Chen, Hongpeng Chen, and Songzhi Su. 2023. [Fine-tuning large language models in education](#). In *Proceedings of the 2023 13th International Conference on Information Technology in Medicine and Education (ITME)*, pages 718–723. IEEE.
- Sankalan Pal Chowdhury, Nico Daheim, Ekaterina Kochmar, Jakub Macina, Donya Rooein, Mrinmaya Sachan, and Shashank Sonkar. 2025. Large language models for education: Understanding the needs of stakeholders, current capabilities and the path forward. *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- Nghia Duong-Trung, Xia Wang, and Miloš Kravčík. 2024. [Bloomllm: Large language models based question generation combining supervised fine-tuning and bloom’s taxonomy](#). In R. Ferreira Mello, N. Rummel, I. Jivet, G. Pishtari, and J. A. Ruipérez Valiente, editors, *Technology Enhanced Learning for Inclusive and Equitable Quality Education*, volume 15160 of *Lecture Notes in Computer Science*. Springer.
- Min Fan, Xinyue Cui, Wanqing Ma, Haiyan Li, Xin Tong, Lin Yang, and Yonghui Wang. 2025. From words to wonder: Designing and evaluating an ai-empowered creative storytelling system for elementary children. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–15.
- Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339.
- Lin Gao, Jing Lu, Zekai Shao, Ziyue Lin, Shengbin Yue, Chiokit Leong, Yi Sun, Rory James Zauner, Zhongyu Wei, and Siming Chen. 2025. [Fine-tuned large language model for visualization system: A study on self-regulated learning in education](#). *IEEE Transactions on Visualization and Computer Graphics*, 31(1):514–524.
- Dominik Glandorf and Detmar Meurers. 2024. Towards fine-grained pedagogical control over english grammar complexity in educational text generation. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 299–308.
- Ariel Han and Zhenyao Cai. 2023. Design implications of generative ai systems for visual storytelling for young learners. In *Proceedings of the 22nd annual ACM interaction design and children conference*, pages 470–474.
- Laura Hanu and Unitary team. 2020. Detoxify. Github. <https://github.com/unitaryai/detoxify>.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text de-generation. In *International Conference on Learning Representations (ICLR)*.
- Chandra Irugalbandara, Ashish Mahendra, Roland Daynauth, Tharuka Kasthuri Arachchige, Jayanaka Dantanarayana, Krisztian Flautner, Lingjia Tang, Yiping Kang, and Jason Mars. 2024. Scaling down to scale up: A cost-benefit analysis of replacing openai’s llm with open source slms in production. In *2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 280–291. IEEE.
- Qiao Jin and Ye Yuan. 2025. “they all look mad with each other”: Understanding the needs and preferences of children and parents in ai-generated images for stories. *International Journal of Child-Computer Interaction*, page 100787.
- Hyeongdon Kim et al. 2025. Readctrl: Personalizing text generation with readability-controlled instruction learning. In *Proceedings of the 2025 Workshop on Intelligent and Interactive Writing Assistant (In2Writing)*, pages 1–15.
- Holly B Lane and Valentina Contesse. 2022. *UFLI foundations: An explicit and systematic phonics program*. Ventris Learning.
- Walter L Leite, Stephanie Hammerschmidt-Snidarich, A Corinne Huggins-Manley, Hatice K Karakis, Holly Lane, Seyedahmad Rahimi, Qian Shen, Xinyi Tai, and Matthew Schmidt. 2025. Storiza: A platform to support children’s oral reading fluency development with generative ai. In *Proceedings of the 19th International Conference of the Learning Sciences-ICLS 2025*, pp. 1574-1578. International Society of the Learning Sciences.
- Jasmine Lesner, Luke Murayama, Tony Guizar, Poom Phunjamaneechot, and Daniel Shapiro. 2024. Ai personalized interactive fiction for young children. In *ECAI 2024*, pages 4756–4763. IOS Press.
- Mengzhe Liao. 2025. Ai-powered storytelling relay: Designing a creative and interactive game for children and parents.
- Zhengyuan Liu, Stella Xin Yin, Dion Hoe-Lian Goh, and Nancy Chen. 2025. [COGENT: A curriculum-oriented framework for generating grade-appropriate](#)

- educational content. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 129–143, Vienna, Austria. Association for Computational Linguistics.
- Tuula Merisuo-Storm and Marjaana Soininen. 2014. Interesting reading materials and exercises encourage also reluctant boys to read. *Procedia-Social and Behavioral Sciences*, 116:2583–2588.
- Subhojyoti Mukherjee, Viet Dac Lai, Raghavendra Ad-danki, Ryan A Rossi, Seunghyun Yoon, Trung Bui, Anup Rao, Jayakumar Subramanian, and Branislav Kveton. 2025. Offline rl by reward-weighted fine-tuning for conversation optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- J Peters. 2006. Reinforcement learning by reward-weighted regression. In *NIPS 2006 Workshop: Towards a New Reinforcement Learning?*
- Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *Advances in Neural Information Processing Systems*, 37:55249–55285.
- Timothy V Rasinski, William H Rupley, David D Pagie, and William Dee Nichols. 2016. Alternative text types to improve reading fluency for competent to struggling readers. *International Journal of Instruction*, 9(1):163–178.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392.
- Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pages 29971–30004. PMLR.
- George Spache. 1953. A new readability formula for primary-grade reading materials. *The Elementary School Journal*, 53(7):410–413.
- Louise Spear-Swerling. 2006. Children’s reading comprehension and oral reading fluency in easy text. *Reading and Writing*, 19(2):199–220.
- Yuling Sun, Jiaju Chen, Bingsheng Yao, Jiali Liu, Dakuo Wang, Xiaojuan Ma, Yuxuan Lu, Ying Xu, and Liang He. 2024. Exploring parent’s needs for children-centered ai to support preschoolers’ interactive storytelling and reading activities. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2):1–25.
- Nathalie J Veenendaal, Margriet A Groen, and Ludo Verhoeven. 2015. What oral text reading fluency can reveal about reading comprehension. *Journal of Research in Reading*, 38(3):213–225.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.
- Shenao Zhang, Zhihan Liu, Boyi Liu, Yufeng Zhang, Yingxiang Yang, Yongfei Liu, Liyu Chen, Tao Sun, and Zhaoran Wang. 2025. Reward-augmented data enhances direct preference alignment of llms. In *International Conference on Machine Learning*, pages 75591–75614. PMLR.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.
- Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texus: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Appendix A: Mathematical Formulas for Evaluation Metrics

Spache Readability score:

This metric estimates the grade level of a text by combining the average sentence length and the percentage of unfamiliar words according to the following formula:

$$Spache = 0.141 \times \bar{L}_{sent} + 0.086 \times P_{unfamiliar} + 0.839 \quad (1)$$

LM-PPL:

For a sequence of tokens $X = (x_1, \dots, x_T)$, this is defined as:

$$PPL(X) = \exp \left(-\frac{1}{T} \sum_{i=1}^T \log p_{\theta}(x_i | x_{<i}) \right) \quad (2)$$

Coherence Score:

Let $E(S_i)$ be the set of named entities in the i -th sentence; the coherence score is derived as:

$$Coh = \frac{1}{N-1} \sum_{i=1}^{N-1} |E(S_i) \cap E(S_{i+1})| \quad (3)$$

where N is the total number of sentences and $|\cdot|$ denotes the cardinality of the set.

Syntactic Complexity:

$$Syn = \frac{1}{N} \sum_{i=1}^N (MDD_i + NSC_i) \quad (4)$$

where N is the total number of sentences, MDD_i is the maximum dependency distance within the i -th sentence (i.e., the largest syntactic distance between a word and its head across all dependency arcs in that sentence), and NSC_i is the number of subordinate clauses in the i -th sentence.

Self-BLEU:

$$\text{Self-BLEU}(s_i) = \text{BLEU}(s_i, \{s_j \mid j \neq i, s_j \in L\}) \quad (5)$$

where s_i denotes the i -th story, L is the set of all stories belonging to the same lesson, and the remaining stories $\{s_j \mid j \neq i, s_j \in L\}$ serve as references for computing the BLEU score.

Repetition between 1,290 stories:

$$\text{GlobalSelfBLEU} = \frac{1}{N} \sum_{i=1}^N \text{BLEU}(s_i, \{s_j \mid j \neq i\}) \quad (6)$$

where N is the total number of stories generated by a given model in a given experiment, and each story s_i is evaluated against all other stories $\{s_j \mid j \neq i\}$ as references.

Appendix B: Tables and Figures

In this appendix, we provide additional details regarding our experimental setups, evaluation results of newly generated stories, examples of the K-2 English reading curriculum.

Table 2: Evaluation results of stories generated by LLMs for all the experiments, $\uparrow$ indicates higher is better; $\downarrow$ indicates lower is better.

Metric	Statistics	Original		Baseline		Good Stories		Rewarded SFT		SFT with Simulated Errors		
		Llama 3.3	GPT-4o	Llama 3	Granite 3.3	Apertus	Llama 3	Granite 3.3	Apertus	Llama 3	Granite 3.3	Apertus
Coherence $\uparrow$	Mean	0.13	0.12	0.02	0.07	0.09	0.11	0.03	0.09	0.12	0.18	0.13
	SD	0.18	0.15	0.05	0.05	0.13	0.09	0.09	0.13	0.14	0.18	0.15
Syntactic Complexity $\downarrow$	Mean	2.81	3.94	4.63	3.72	3.41	4.30	3.94	3.14	3.38	3.12	2.96
	SD	1.08	1.03	2.02	1.05	0.90	1.49	1.09	0.95	1.11	0.88	0.98
Spache Readability $\downarrow$	Mean	2.54	3.31	4.05	3.52	2.83	3.38	3.38	2.63	2.71	2.56	2.34
	SD	0.79	0.85	1.04	0.93	0.47	1.26	0.62	0.54	0.70	0.60	0.68
Toxicity $\downarrow$	Mean	0.10	0.00	0.01	0.06	0.00	0.01	0.00	0.01	0.01	0.02	0.02
	SD	0.04	0.02	0.04	0.12	0.01	0.03	0.01	0.04	0.02	0.06	0.07
LM-PPL $\downarrow$	Mean	26.71	28.08	23.16	24.91	16.49	21.86	23.27	23.64	16.86	14.55	19.73
	SD	10.95	8.57	13.22	10.21	3.88	10.09	8.77	10.71	4.29	3.56	6.64
Repetition in lessons $\downarrow$	Mean	0.62	0.09	0.03	0.11	0.12	0.03	0.04	0.06	0.11	0.12	0.09
	SD	0.16	0.05	0.03	0.09	0.06	0.03	0.03	0.04	0.06	0.07	0.08
Total repetition $\downarrow$	Mean	0.42	0.19	0.21	0.37	0.40	0.19	0.27	0.31	0.37	0.42	0.32
	SD	0.20	0.06	0.09	0.16	0.08	0.08	0.07	0.08	0.10	0.11	0.13
											0.05	0.10
											0.21	0.27
											0.11	0.11

A "bad" story and a "good" story

A bad case: The bustling town of Eldoria was known for its legendary festival, where every year, children from far and wide gathered to witness the magical spectacle that unfolded in the heart of the forest. This time, the theme was Whimsical Wonders, and the excitement was palpable as everyone prepared for the grand event. Little Lily, a bright-eyed girl with an insatiable curiosity, had always dreamt of attending this festival. She had heard tales of towered castles, explored hidden paths, and even murmured secrets whispered by the trees themselves. With her trusty backpack filled with snacks and a notebook to jot down her adventures, she set off on the journey. As Lily ventured deeper into the enchanting woods, she came across a peculiar sight. Fern towered whirlwind unbreakable.

A good case: Pam and Pat are going to see a monkey. The kids got up at dawn. Pam and Pat put on their pants and grab a banana. The sun is up, and the kids are ready to go. Pam and Pat jump on a bus to head downtown. "I can not wait to see the monkey," said Pam. "Me too!" said Pat. The kids sit on a bench and eat a snack. A man hands Pam a map. The kids look at the map and find the path to the zoo. Pam and Pat walk up to the monkey and hand it a banana. The monkey eats the banana and jumps around. Pam and Pat laugh and clap. The monkey is fun to watch. Pam and Pat say goodbye to the monkey and get on the bus. "That was a great trip," said Pam. "I had a blast," said Pat. The kids go home and tell their mom about the monkey.

Figure 2: A qualitative example of a bad case and a good case in our stories.

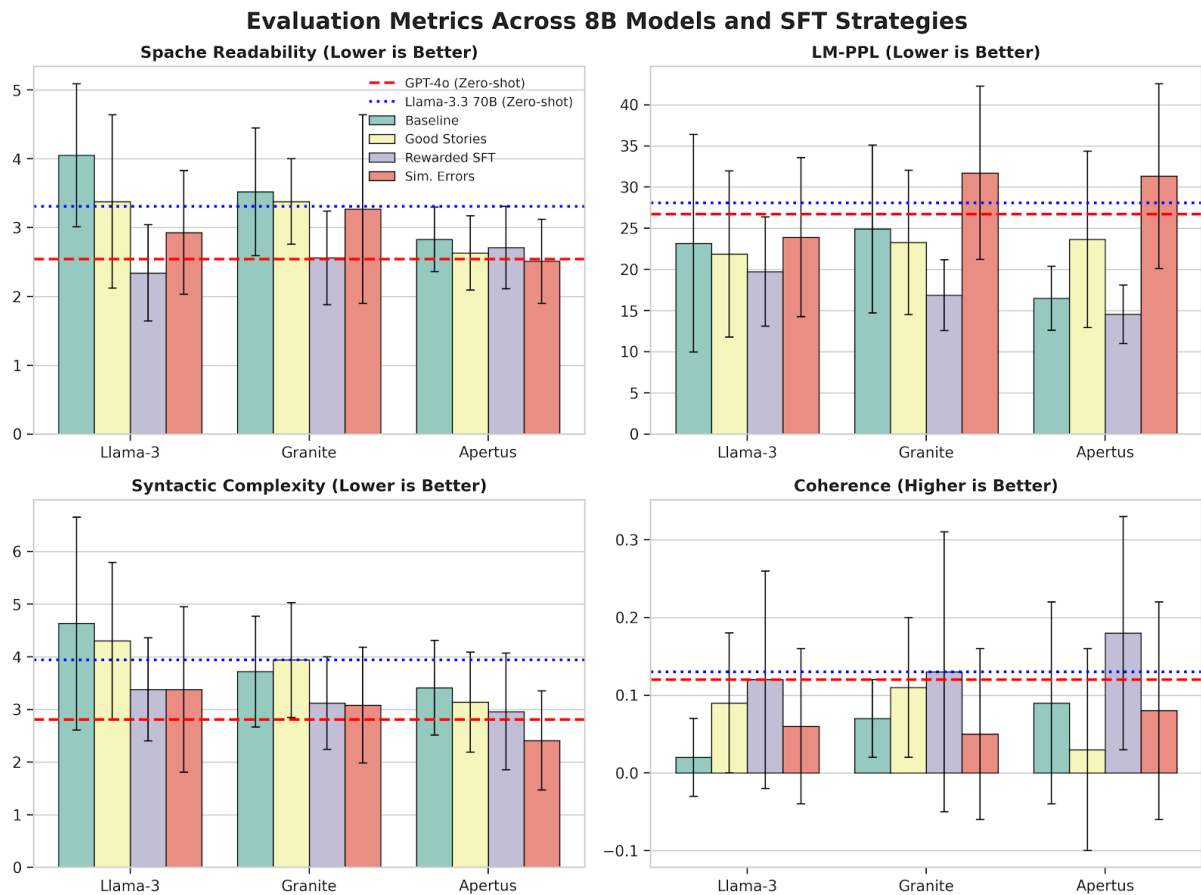


Figure 3: Evaluation metrics across 8B models and SFT strategies. We compare our four fine-tuning methods against the zero-shot performance of GPT-4o (red dashed line) and Llama-3.3 70B (blue dotted line). Error bars represent the standard deviation (SD). Lower scores indicate better performance for Spache Readability, LM-PPL, and Syntactic Complexity, while higher scores are better for Coherence. The Rewarded SFT strategy consistently drives the 8B models to approximate or surpass the massive baseline models.

K–2 English Reading Curriculum

Lesson ID: 30

Grade: Kindergarten

Phonemes & Constraints: a, m, s, t, cvc words, p, f, i, n, cvc words, nasalized, a, o, d, c, u, g, b, e, cvc words, s, k, h, r, l, w, j, y

The Story of Experts:

A yak looks at a plant. It’s a yam plant. The yak digs to get a yam from the plant. The yak stomps on the yam to cut it into bits. A dog sees the plant. The dog digs to get a yam from the plant. The dog can not stomp on the yam. The yak helps the dog. “Yum,” said the yak, “yams are the best!” “Yes,” said the dog, “yum, yum, yum!”

Figure 4: An example of the lessons in the K–2 English reading curriculum we used.

Transformer-based readability classifiers are worse than you think: Evidence from cross-domain Arabic readability assessment

Sarah Alzubi

Brigham Young University
sarah.alzubi01@gmail.com

Robert Reynolds

Brigham Young University
robert_reynolds@byu.edu

Abstract

Arabic readability assessment is under-explored compared to English, and existing models are typically evaluated only within the training domain. We introduce the Jordanian School Textbook Corpus (JSTC), 82,512 segments from 240 textbooks spanning grades 1–12, and combine it with DARES to train XGBoost classifiers, fine-tuned CAMELBERT transformers, and hybrid architectures evaluated both in-domain and on the BAREC out-of-domain benchmark. CAMELBERT achieves strong in-domain performance (QWK = 0.830) but its cross-domain QWK collapses to 0.085, while XGBoost over 127 handcrafted linguistic features alone maintains the highest cross-domain QWK (0.240); adding [CLS] embeddings to those features actively harms transfer. Probing reveals that CAMELBERT layers implicitly capture some linguistic features but higher-level signals overwhelm them, and Captum attribution identifies nouns and nominal particles such as *al-* as the most important tokens. The results argue for prioritizing linguistically-grounded features over contextual embeddings when cross-domain robustness is required.

1 Introduction

Readability assessment—the task of estimating the difficulty level of a text for a target audience—is fundamental to educational applications such as textbook selection, curriculum design, and adaptive learning systems (Vajjala, 2022).¹ While substantial progress has been made for English, Arabic readability research remains comparatively under-resourced despite Arabic being a major world language spoken by over 400 million people (Cavalli-Sforza et al., 2018). Arabic presents unique chal-

lenges for readability assessment that cannot be addressed by simply transferring English-centric methods. Its morphological system is exceptionally rich: a single Arabic word can encode subject agreement, definiteness, case, mood, and multiple clitics, making morphosyntactic features far more informative for readability than in morphologically simpler languages. Orthographic variation, optional diacritics, and the diglossia between Modern Standard Arabic (MSA) and regional dialects further complicate both corpus construction and feature extraction.

Recent work in readability assessment has followed two broad paradigms. The first employs handcrafted linguistic features—lexical diversity measures, syntactic complexity metrics, and morphological statistics—fed to traditional classifiers (Deutsch et al., 2020; Al-Tamimi et al., 2014; El-Haj and Rayson, 2016). The second fine-tunes pretrained transformer models such as BERT on labeled readability data, using the [CLS] token representation as input to a classification head (Lee et al., 2021; Imperial, 2021; Ouassil et al., 2024). Hybrid approaches that combine both feature types have also been explored, with mixed results (Lee et al., 2021). For Arabic specifically, the transformer paradigm has gained traction with the release of pretrained Arabic language models such as CAMELBERT (Inoue et al., 2021), enabling direct fine-tuning for downstream tasks.

A critical limitation of existing work, however, is that models are almost universally evaluated only in-domain—that is, on held-out data from the same distribution as the training set. This evaluation protocol can mask a fundamental problem: neural models may learn to exploit surface-level correlations and semantic peculiarities specific to the training corpus rather than capturing genuine text complexity. Practical educational deployment requires readability models that generalize across text types: a model trained on textbooks must also work

¹This article is an abbreviated version of Alzubi (2026). Code and data are available in a private repo (due to copyright restrictions) at https://github.com/Sarah-git99/2026_BEAArabicReadability.git. Contact the authors for access.

reliably for news articles, literature, and online content encountered by students. This cross-domain robustness has received almost no systematic attention for Arabic readability, and recent evidence from other languages suggests that neural models may fail dramatically under domain or format shift (Van Der Sluis and Van Den Broek, 2025; Lim et al., 2024).

The stakes of this question are not merely academic. As readability assessment tools are adopted in educational settings—to match learners with appropriate materials, to evaluate textbook difficulty across curricula, and to support adaptive learning platforms (Vajjala, 2022)—their cross-domain reliability becomes a practical concern. If these tools rely on models that perform well on benchmarks but fail in real-world deployment across diverse text types, the consequences could include mismatched materials, frustrated learners, and misallocated educational resources.

In this paper, we present the first systematic comparison of linguistic, neural, and hybrid readability models for Arabic under both in-domain and cross-domain conditions. Our contributions are:

1. **Jordanian School Textbook Corpus (JSTC):** A new corpus of approximately 82,512 text segments from 240 Jordanian school textbooks (grades 1–12), with 127 extracted linguistic features across six categories.
2. **Cross-domain evaluation:** A rigorous comparison of seven model configurations on both 1) withheld in-domain textbook data and 2) the BAREC benchmark (Elmadani et al., 2025), revealing that fine-tuned transformers suffer catastrophic performance collapse under domain shift.
3. **Key finding:** XGBoost with linguistic features alone achieves the highest cross-domain Quadratic Weighted Kappa (QWK), substantially outperforming all transformer and hybrid configurations. Adding [CLS] embeddings to linguistic features actively degrades cross-domain performance.

2 Related Work

2.1 Arabic Readability Corpora and Methods

Early Arabic readability work adapted traditional formulas to Arabic’s morphological properties. Al-Tamimi et al. (2014) proposed AARI, an automatic readability index for Arabic, and El-Haj and

Rayson (2016) introduced OSMAN, a readability metric tailored to Arabic text. Forsyth (2014) applied statistical machine-learning methods for automatic readability detection for Modern Standard Arabic. Cavalli-Sforza et al. (2018) surveyed the state of Arabic readability research, identifying corpus scarcity as a major bottleneck.

More recently, several Arabic readability corpora have been developed. Nassiri et al. (2019) introduced MoSAR, a Modern Standard Arabic readability corpus for L1 learners. El-Haj et al. (2024) and Almujaivel et al. (2025) released DARES, a dataset for Arabic readability estimation derived from Saudi school materials. Most recently, Elmadani et al. (2025) introduced BAREC, a large-scale benchmark spanning multiple domains and grade levels. Despite this progress, none of these efforts has systematically evaluated cross-domain generalization of their models.

2.2 Transformer and Hybrid Approaches

Lee et al. (2021) demonstrated that combining BERT soft labels with handcrafted linguistic features improves readability assessment for English, achieving state-of-the-art results on the CommonLit benchmark. Imperial (2021) showed that BERT embeddings alone can be effective for readability classification. For Arabic specifically, Ouassil et al. (2024) combined BERT with BiLSTM for readability assessment, and Khallaf and Sharoff (2021) applied both sentence embeddings and traditional linguistic features for readability classification.

2.3 Cross-Domain Robustness

Cross-domain evaluation of readability models has received limited but growing attention. Van Der Sluis and Van Den Broek (2025) showed that interpretable models exhibit better domain generalization in textual complexity modeling across multiple languages, providing theoretical motivation for feature-based approaches. Their analysis demonstrates that models with transparent, linguistically-grounded features avoid the domain-specific overfitting that characterizes opaque neural representations. Lim et al. (2024) investigated the robustness of hybrid models in cross-domain readability assessment, finding that linguistic features provide more stable transfer than purely neural representations. Deutsch et al. (2020) provided a thorough evaluation of linguistic features for English readability at the BEA workshop, establishing that traditional features remain competitive with neural

approaches, though they did not conduct cross-domain experiments.

To our knowledge, no prior study has systematically compared linguistic, neural, and hybrid readability models for Arabic under cross-domain conditions. Our work fills this gap, leveraging Arabic’s rich morphological system to construct a particularly informative linguistic feature set and using the recently released BAREC benchmark to perform rigorous out-of-domain evaluation.

3 Data

3.1 Jordanian School Textbook Corpus (JSTC)

We construct the Jordanian School Textbook Corpus (JSTC) from 240 official textbooks in the Jordanian national curriculum, spanning Grades 1–12. These textbooks cover core subjects including Arabic Language, Islamic Education, and Digital Skills.

Text extraction from the textbook PDFs presented a significant challenge due to the complexity of Arabic script, including right-to-left writing, diacritical marks, complex ligatures, and mixed text-image layouts common in educational materials. Standard PDF text extraction tools produced unreliable output for Arabic content, so we employed the QARI-OCR model (Wasfy et al., 2025) for advanced character-level recognition of Arabic text from scanned pages. This OCR-based pipeline was essential for handling the typographic diversity of textbooks, which include tables, sidebars, and decorative elements alongside running text.

Preprocessing was performed using CAMEL Tools (Obeid et al., 2020). Text was dediacritized to remove optional vowel markings that are inconsistently applied across textbooks, and letters were normalized (e.g., variant forms of *alif* and *ya*’ were mapped to canonical forms) to reduce orthographic variation. Deduplication was applied to remove repeated passages appearing across multiple textbooks in the same curriculum (e.g., recurring introductory material); this process removed 5.42% of segments from JSTC. The resulting corpus comprises 82,512 text segments totaling over 2.1 million tokens. Segments preserve the natural textbook structure, including paragraphs, passages, bullet points, and standalone sentences, rather than being artificially re-segmented into uniform units.

Table 1 shows the distribution of segments and tokens across 12 grades, with lines between the

Coarse label	Grade	Segments	Tokens
L1	G1	2263	36063
	G2	3188	51103
	G3	3042	55950
L2	G4	6042	108681
	G5	5341	107208
	G6	4577	105726
L3	G7	7991	225743
	G8	7356	212545
	G9	10012	273309
L4	G10	16677	572634
	G11	15604	483499
	G12	11172	361304
Total		82512	1764899

Table 1: Distribution of segments and tokens in JSTC dataset.

four coarse-grained readability categories. The corpus exhibits substantial class imbalance, with high school grades contributing roughly five times as many segments as early elementary grades, reflecting the greater volume of advanced instructional material typical of readability corpora.

3.2 DARES

To introduce curricular diversity into training, we supplement JSTC with DARES V1 (El-Haj et al., 2024), a readability corpus derived from the Saudi Arabian public school curriculum, grades 1–12. After deduplication (19.37% of DARES segments were removed), the combined training set comprises approximately 93,265 segments.

3.3 BAREC Cross-Domain Evaluation

For cross-domain evaluation, we use BAREC (Elmadani et al., 2025), a sentence-level multi-domain Arabic readability benchmark spanning three broad domains: Arts & Humanities, Social Sciences, and STEM. BAREC was used exclusively for cross-domain testing—none of its data was observed during training. This enables us to assess whether models capture generalizable readability signals or merely overfit to textbook-specific patterns. Both 12-class (individual grades) and 4-class (educational bands) label mappings are applied to BAREC for evaluation.

Evaluating on BAREC provides a particularly stringent test of cross-domain robustness because the BAREC texts differ from textbook prose along multiple dimensions simultaneously: topic, genre, register, text length, and intended audience.

A model that maintains predictive accuracy on BAREC demonstrates genuinely transferable readability representations.

4 Features and Models

We extract 127 handcrafted linguistic features from each text segment, organized into six categories designed to capture multiple dimensions of text complexity:

Lexical features include type-token ratio (TTR), moving-average TTR (MATTR; [Covington and McFall, 2010](#)), measure of textual lexical diversity (MTLD; [McCarthy and Jarvis, 2010](#)), average word length, and lexical density (ratio of content words to total words).

TF-IDF features comprise eight aggregated statistics inspired by [Attia et al. \(2023\)](#) computed from both surface forms and lemmatized forms, capturing rates of lexical distinctiveness within the corpus.

Syntactic features measure clause density, clause embedding depth, sentence length, and dependency parse structure using CamelParser 2.0 ([Elshabrawy et al., 2023](#)) with Universal Dependencies annotation.

Part-of-speech features capture the distribution of POS categories (nouns, verbs, adjectives, adverbs, pronouns, prepositions, and functional categories) as proportions within each segment.

Morphosyntactic features exploit Arabic’s rich morphological system, including person, number, and gender agreement; definiteness marking (the *al*-prefix); verb mood and tense; and case marking—all extracted via CAMEL Tools ([Obeid et al., 2020](#)) using the D3TOK tokenization scheme.

Semantic features include Latent Dirichlet Allocation (LDA) topic distributions at multiple granularities ($K \in \{50, 100, 150, 200\}$) and derived measures of semantic richness, clarity, focus, and breadth, along with word frequency quartile analysis that captures lexical sophistication.

Together, these 127 features are designed to capture readability signals at multiple linguistic levels—from surface lexical statistics through syntactic structure to semantic content—while exploiting Arabic’s rich morphological system in ways that generic transformer representations may not.

4.1 Models

We evaluate seven model configurations spanning statistical, neural, and hybrid approaches. Of the many statistical classifiers tested, we report results for XGBoost, which consistently outperformed alternatives such as logistic regression, SVM, and Random Forests in preliminary experiments.

XGBoost-Ling: XGBoost ([Chen and Guestrin, 2016](#)) trained on the 127 linguistic features.

XGBoost-CLS: XGBoost trained on the 768-dimensional [CLS] embedding extracted from a frozen CAMELBERT model.

XGBoost-Ling+CLS: XGBoost trained on the concatenation of linguistic features and [CLS] embeddings.

CAMELBERT-MSA: CAMELBERT ([Inoue et al., 2021](#)) pretrained on Modern Standard Arabic, fine-tuned on the readability task.

CAMELBERT-MIX: CAMELBERT pretrained on a larger mixed-dialect corpus, fine-tuned on the readability task.

CAMELBERT-MSA+H1: CAMELBERT-MSA with the H1 subset of linguistic features (features which had very low correlation with probing model outputs) projected and fused via concatenation before the classification head.

CAMELBERT-MIX+H3: Same architecture as CAMELBERT-MSA+H1, but using CAMELBERT-MIX and the H3 set of linguistic features (features which had very low, low, and moderate correlation with probing model outputs).

4.2 Experimental Setup

All models were trained on the combined JSTC and DARES data with a standard train/validation/test split. In-domain evaluation was performed on a held-out textbook test set drawn from the same corpus; cross-domain evaluation was performed on the full BAREC corpus, which was not observed during training. This strict separation ensures that cross-domain results reflect genuine generalization rather than data leakage.

We evaluate at two granularities: a **fine-grained** task with 12 classes (individual grade levels G1–G12) and a **coarse-grained** task with 4 classes (the educational bands in Table 1). The fine-grained task is particularly challenging because adjacent grade levels may differ only subtly in linguistic complexity, while the coarse-grained task groups these levels into broader educational stages.

For XGBoost models, we use the XGBoost

gradient-boosted tree classifier (Chen and Guestrin, 2016) with hyperparameters selected via cross-validation on the training set: `n_estimators=600`, `learning_rate=0.03`, `max_depth=6`. XGBoost is well-suited to this task because its tree-based architecture can capture non-linear interactions between linguistic features (e.g., the combination of high lexical density and long sentence length may indicate a different difficulty level than either feature alone).

For CAMELBERT models, we fine-tune from the pretrained checkpoints with a classification head on top of the [CLS] representation. CAMELBERT-MSA was pretrained on Modern Standard Arabic text, while CAMELBERT-MIX was pretrained on a larger corpus that includes dialectal Arabic, potentially providing broader linguistic coverage at the cost of MSA specificity.

The hybrid models employ a projection fusion mechanism: the 127-dimensional linguistic feature vector is projected via a learned linear layer to match the 768-dimensional [CLS] representation. The projected features are then concatenated with the [CLS] output and passed to the classification head. This architecture allows the model to learn to weight both signal sources during training, but—as our results show—tends to prioritize the higher-dimensional [CLS] component. The models were trained using a maximum sequence length of 512, a learning rate of 5×10^{-5} , a batch size of 32, and 10 epochs. All experiments were conducted using 5-fold cross-validation.

Performance is measured using accuracy and Quadratic Weighted Kappa (QWK; Cohen, 1968). QWK is a variant of Cohen’s Kappa that applies quadratic penalties to prediction errors proportional to their distance from the true label, making it particularly appropriate for ordinal classification where a prediction of grade 3 for a grade 4 text is less severe than a prediction of grade 1. QWK ranges from -1 (systematic disagreement) through 0 (chance agreement) to 1 (perfect agreement). We adopt QWK as our primary evaluation metric, supplemented by accuracy.

5 Results

5.1 Fine-Grained Results (12 Classes)

Table 2 presents the fine-grained results. In-domain, CAMELBERT-MSA achieves the highest performance across all metrics (Acc = 0.640, QWK = 0.830), substantially outperforming XGBoost-Ling

Model	Ling	CLS	Acc (In)	QWK (In)	Acc (Cross)	QWK (Cross)
XGBoost-Ling	✓		0.284	0.460	0.130	0.240
CAMELBERT-MSA		✓	0.640	0.830	0.098	0.085
CAMELBERT-MIX		✓	0.626	0.807	0.108	0.092
XGBoost-CLS		✓	0.380	0.643	0.091	0.161
CAMELBERT-MSA+H1	✓	✓	0.644	0.821	0.101	0.089
CAMELBERT-MIX+H3	✓	✓	0.630	0.810	0.110	0.077
XGBoost-Ling+CLS	✓	✓	0.390	0.670	0.080	0.170

Table 2: Fine-grained (12-class) readability results. Best in-domain scores in **bold** (top section); best cross-domain scores in **bold** (all rows). XGBoost with linguistic features alone achieves the highest cross-domain QWK.

(Acc = 0.284, QWK = 0.460). The hybrid model CAMELBERT-MSA+H1 provides only marginal improvements in accuracy, but is slightly worse than its base in QWK (0.821 vs. 0.830).

The cross-domain results tell a strikingly different story. CAMELBERT-MSA’s QWK collapses from 0.830 to 0.085—a 90% relative decrease—and accuracy drops from 0.640 to 0.098, approaching the chance level of 0.083 for 12 classes. All models that rely on [CLS] representations show similarly catastrophic degradation. In contrast, **XGBoost-Ling achieves the highest cross-domain QWK at 0.240**, more than double that of any CLS-based model.

Notably, adding [CLS] embeddings to linguistic features (XGBoost-Ling+CLS) *reduces* cross-domain QWK from 0.240 to 0.170, indicating that contextual embeddings actively introduce domain-specific noise that harms transfer. The same pattern holds for accuracy: XGBoost-Ling (0.130) outperforms XGBoost-Ling+CLS (0.080) cross-domain, despite the latter’s higher in-domain accuracy (0.390 vs. 0.284).

5.2 Coarse-Grained Results (4 Classes)

Table 3 shows the coarse-grained results, which reinforce the same pattern. In-domain, CAMELBERT-MSA again dominates (Acc = 0.776, QWK = 0.802), with the hybrid H1 providing a slight improvement (QWK = 0.805).

Cross-domain, the pattern of transformer collapse persists. CAMELBERT-MSA’s QWK drops from 0.802 to 0.106, while XGBoost-Ling’s QWK drops more modestly from 0.414 to 0.264. **XGBoost-Ling again achieves the highest cross-domain QWK (0.264)**, outperforming the best hybrid model by a factor of three (H1: 0.084).

Even more strikingly, XGBoost-Ling achieves the highest cross-domain accuracy (0.470), surpassing all transformer and hybrid models despite

Model	Ling	CLS	Acc (In)	QWK (In)	Acc (Cross)	QWK (Cross)
XGBoost-Ling	✓		0.550	0.414	0.470	0.264
CAMeLBER-MSA		✓	0.776	0.802	0.373	0.106
CAMeLBER-MIX		✓	0.765	0.785	0.352	0.051
XGBoost-CLS		✓	0.612	0.600	0.360	0.160
CAMeLBER-MSA+H1	✓	✓	0.780	0.805	0.363	0.069
CAMeLBER-MIX+H3	✓	✓	0.770	0.793	0.370	0.067
XGBoost-Ling+CLS	✓	✓	0.630	0.630	0.320	0.155

Table 3: Coarse-grained (4-class) readability results. Best in-domain scores in **bold** (top section); best cross-domain scores in **bold** (all rows). XGBoost with linguistic features alone achieves the highest cross-domain QWK.

its much lower in-domain accuracy. This reversal highlights the fundamental difference between in-domain pattern matching and cross-domain generalization.

The coarse-grained results also reveal an interesting asymmetry in the hybrid models. While H1 achieves the highest in-domain QWK (0.805), its cross-domain QWK (0.084) is actually *lower* than that of standalone CAMeLBER-MSA (0.103). This suggests that the projection-fusion mechanism, while effective at combining signals for in-domain classification, may amplify overfitting to domain-specific patterns when cross-domain transfer is required. The same trend holds for H3 (cross-domain QWK 0.067 vs. CAMeLBER-MIX’s 0.060), though the difference is smaller.

5.3 Cross-Domain Subdomain Analysis

Performance on the BAREC subdomains varies substantially by topic area. For the coarse-grained task, XGBoost-Ling achieves the best cross-domain results on STEM texts (Acc = 0.679, QWK = 0.293) and Social Sciences (Acc = 0.617, QWK = 0.287), with somewhat lower performance on Arts & Humanities (Acc = 0.427, QWK = 0.239). CAMeLBER’s cross-domain performance remains near chance across all subdomains, confirming that the transformer’s domain-specific overfitting is not limited to a single target domain.

For the fine-grained task, subdomain variation follows a similar pattern. XGBoost-Ling achieves cross-domain QWK of 0.260 on STEM, 0.220 on Social Sciences, and 0.210 on Arts & Humanities. CAMeLBER-MSA shows uniformly poor cross-domain performance across all subdomains (QWK < 0.10).

This pattern suggests that linguistic complexity features transfer more reliably to STEM and Social Science texts, which share certain structural properties with textbook prose (e.g., expository structure,

technical vocabulary, formal register). Literary and humanities texts present a greater distributional shift due to their use of narrative structures, figurative language, and stylistic variation that differ markedly from instructional prose.

5.4 Feature Type Comparison

The XGBoost variants isolate the effect of input representation under a constant classifier. In-domain, the QWK ordering is Ling < CLS < Ling+CLS for both the fine-grained task (0.460, 0.643, 0.670) and the coarse-grained task (0.414, 0.600, 0.630). Cross-domain, this ordering reverses: Ling > Ling+CLS ≥ CLS (fine-grained: 0.240, 0.170, 0.160; coarse-grained: 0.264, 0.160, 0.160). The near-equality of CLS and Ling+CLS cross-domain suggests that when both feature types are present, the classifier relies primarily on CLS features—which dominate in-domain—effectively discarding the linguistic features that would have provided better transfer. **The features most useful for in-domain classification are precisely those that hurt cross-domain transfer**, an inherent tension that cannot be resolved by naive feature combination.

5.5 Feature Importance

We performed permutation importance analysis for the XGBoost model on the test set. This method evaluates feature importance by randomly shuffling one feature at a time and measuring the resulting drop in model performance. Table 4 presents the top 6 ranked feature importances based on the observed performance degradation. The two rankings diverge sharply: in-domain predictions rely most heavily on surface lexical and TF-IDF statistics (lemma_category_tfidf_mean, form_category_tfidf_mean, avg_word_length, Digit_Density), whereas cross-domain predictions depend predominantly on syntactic complexity (avg_parse_tree_height, avg_sentence_length_morphemes, avg_sentence_length_words) and basic vocabulary coverage—features that capture domain-independent structural properties of text complexity rather than corpus-specific lexical distributions, reinforcing the pattern observed in §5.4.

5.6 Probing Analysis

Figure 1 presents the distribution of the R^2 correlations of linguistic features across layers. Specifi-

Feature	In-domain	Cross-domain
lemma_category_tfidf_mean	0.0414	—
n_sentences	0.0258	—
per_1	0.0248	—
form_category_tfidf_mean	0.0239	—
avg_word_length	0.0200	—
Digit_Density	0.0189	—
avg_parse_tree_height	—	0.0078
avg_sentence_length_morphemes	—	0.0049
avg_sentence_length_words	—	0.0041
per_3	—	0.0035
Basic_Vocab_Ratio	—	0.0028
pos_pron_dem	—	0.0028

Table 4: Top 6 permutation feature importance for in-domain and cross-domain test sets.

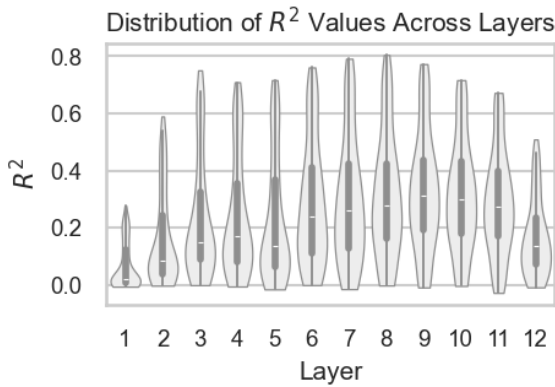


Figure 1: Distribution of probe R^2 values across CAMELBERT layers for linguistic feature groups.

cally, it shows that the lower layers (1–3) encode a limited range of features and only partially capture them, mainly focusing on lexical cues and surface-level information. In contrast, the middle layers show stronger correlations across most linguistic features, indicating more effective encoding in these layers. This suggests that linguistic features are best represented in the middle layers. In the upper layers (10–12), the R^2 values stabilize or decrease slightly. This pattern indicates that, while linguistic information is still present, the representations become more specialized toward semantic abstraction and downstream tasks. See Alzubi (2026) for the full list of feature correlations with each layer.

5.7 Captum Attribution Analysis

We use the Captum attribution analysis to identify the textual elements CAMELBERT-MSA relied on most for readability prediction. The most influential POS categories are nouns and adjectives, with morphological dominance driven by singular num-

ber and masculine gender, and clitic importance led by definiteness and coordination markers. The model assigns the highest importance to content words, especially nouns and proper nouns, which show positive excess attribution across grades, indicating a consistent signal for readability. Verbs, adjectives, and adverbs vary across grades, while function words generally show weaker or negative contributions, with grammatical words increasing at higher grades, suggesting stronger syntactic encoding. For clitics, the definite article *al-* has the strongest attribution across grades, while *wa-* becomes more prominent in higher grades, likely reflecting increased coordination and syntactic complexity.

6 Discussion

6.1 Why Transformers Fail Cross-Domain

The dramatic cross-domain collapse of fine-tuned CAMELBERT models suggests that these models learn to exploit domain-specific topical and stylistic cues present in textbook data rather than genuinely capturing text complexity. The [CLS] representation, while powerful for in-domain classification, appears to encode distributional patterns specific to the training domain—patterns that become misleading when the text distribution shifts.

This interpretation is supported by two additional observations. First, XGBoost-CLS (which uses the same [CLS] embeddings but in a different classifier) also degrades substantially cross-domain, confirming that the problem lies in the representation rather than the classification architecture. Second, adding linguistic features to the transformer (hybrid models H1 and H3) fails to rescue cross-domain performance ($QWK \leq 0.089$ for fine-grained), suggesting that the [CLS] signal dominates the combined representation and overwhelms the more transferable linguistic features.

We note, however, that BAREC differs from the training data not only in topic and register but also in format: BAREC items are single sentences, whereas JSTC and DARES segments are typically multi-sentence passages. Part of the observed collapse may therefore reflect this format shift rather than pure domain shift, particularly for [CLS] representations that may encode passage-level distributional cues unavailable in short sentence inputs.

6.2 Why Linguistic Features Transfer Better

Linguistic features such as average sentence length, lexical diversity (MATTR, MTLTD), clause density, morphological complexity, and POS distributions measure domain-independent properties of text complexity. A syntactically complex sentence remains complex regardless of whether it discusses photosynthesis or Ottoman history. These features thus provide a noisier but more robust signal that maintains predictive value under domain shift.

The finding that XGBoost-Ling+CLS performs *worse* than XGBoost-Ling cross-domain is particularly telling. It demonstrates that [CLS] embeddings do not merely add complementary information—they actively introduce domain-specific noise that degrades the classifier’s ability to exploit the transferable linguistic signal.

The probing results further confirm a layered encoding structure. Lower layers capture surface-level and lexical features, while middle layers are more strongly associated with syntactic information. In contrast, higher layers show reduced sensitivity to explicit linguistic features and instead emphasize task- and domain-specific contextual representations, aligning with the behavior of the classification head.

Several feature categories identified as important for cross-domain transfer are only weakly encoded in the [CLS] representation. Fine-grained POS distributions, rare morphological markers, and lexical frequency statistics show consistently low probe performance across layers. These features capture domain-independent aspects of text complexity, yet remain underrepresented in the learned embedding. This explains strong in-domain results, where topical signals are sufficient, and weaker cross-domain performance, where such signals do not transfer reliably. It also accounts for the limited benefit of hybrid models, since well-encoded features are largely redundant, while weakly encoded ones contribute too little to improve performance.

6.3 Comparison with Prior Work

Our in-domain CAMELBER results (QWK = 0.830 on 12 grades) are competitive with recent Arabic readability benchmarks. Elmadani et al. (2025) report QWK of 0.810 on BAREC’s in-domain evaluation, and Almujaivel et al. (2025) report Macro-F1 of 0.420 on concept-based readability estimation. Our linguistic XGBoost model’s in-domain QWK (0.460) is lower, reflecting the

well-documented advantage of contextual representations for in-domain classification. However, the cross-domain comparison—which no prior Arabic readability study has systematically conducted—reveals that this in-domain advantage is misleading.

Our findings are consistent with Van Der Sluis and Van Den Broek (2025), who demonstrated across multiple languages and tasks that interpretable models generalize better than neural models under domain shift. They extend the English-language findings of Deutsch et al. (2020), who showed that linguistic features remain competitive with neural approaches for readability—though that study did not include cross-domain evaluation. Critically, our results contrast with work that has found hybrid approaches superior to either feature type alone (Lee et al., 2021); we show that this finding does not hold under domain shift, where the hybrid’s neural component becomes a liability rather than an asset.

6.4 Implications for Educational NLP

A readability tool deployed in real educational settings must handle diverse text types: textbooks, supplementary readings, news articles, and online content. Our results demonstrate that current transformer-based approaches, despite their impressive in-domain performance, are not ready for such deployment without explicit domain-adaptation strategies. Linguistic-feature models, while exhibiting lower in-domain performance, provide a substantially safer baseline for cross-domain use.

This finding has particular relevance for Arabic educational NLP, where the diversity of text types, genres, and registers encountered by learners is wide, and where the cost of misclassifying a text’s difficulty level can directly affect curriculum alignment and learner outcomes. A model that correctly classifies textbook passages but assigns university-level STEM texts to the elementary band would be worse than useless in a mixed-resource educational setting.

More broadly, cross-domain evaluation should be a standard component of readability assessment research, particularly when the intended application involves diverse input texts. Although XGBoost-Ling achieved a modest 0.240 QWK, this result is qualitatively superior to the near-chance performance of the neural models we tested.

7 Conclusion

We introduced JSTC, a new Arabic readability corpus consisting of approximately 82,512 textbook segments with 127 linguistic features, which was further combined with the DARES dataset comprising 10,755 samples, and conducted the first systematic cross-domain evaluation of Arabic readability models. Our central finding is that fine-tuned transformers, despite achieving strong in-domain QWK (0.830), collapse to near-chance under domain/format shift (QWK 0.085), while XGBoost with linguistic features alone maintains the most robust cross-domain performance (QWK 0.240–0.264). Adding [CLS] embeddings to linguistic features actively degrades cross-domain transfer.

These results argue for prioritizing linguistically-grounded, interpretable feature sets over opaque contextual representations in educational readability applications, particularly when cross-domain robustness is required. The finding that contextual embeddings actively harm cross-domain transfer—even when combined with linguistic features—suggests that the two feature types encode fundamentally different types of information, and that naive combination strategies are insufficient.

Future work should explore domain adaptation techniques (e.g., adversarial training, domain-invariant representations) that could improve transformer robustness without sacrificing in-domain performance, as well as lighter-weight regularizers such as label smoothing and early stopping that may reduce overconfidence in domain-specific patterns. Evaluation on additional cross-domain benchmarks beyond BAREC—such as the Arabic portion of ReadMe++ (Naous et al., 2024), which provides sentence-level readability labels across multiple domains—would help confirm the generality of our findings and disentangle domain shift from format shift; rotating which of JSTC, DARES, and BAREC serves as the held-out corpus would further isolate corpus-specific effects. More balanced training data, particularly with expanded early-grade coverage, would also help mitigate the imbalance-induced biases observed here. Finally, investigating whether recent larger Arabic language models or instruction-tuned models exhibit improved cross-domain transfer would clarify whether our findings reflect a fundamental limitation of the fine-tuning paradigm or a limitation of the specific model scale studied here.

Limitations

BAREC was selected as our cross-domain test corpus because it represents the most extreme distributional shift available, but it differs from the JSTC and DARES training data along multiple dimensions simultaneously—topic, register, and format. In particular, BAREC items are single sentences while JSTC and DARES segments are typically multi-sentence passages, so part of the observed cross-domain collapse may reflect this format shift rather than pure domain shift.

The set of cross-domain regularization strategies explored in this study was limited, which may have affected the absolute performance reported for neural and hybrid models across datasets and text sources.

Both the in-domain and cross-domain datasets are imbalanced, particularly across subdomains and early grade levels, which may have influenced model training and evaluation.

Ethics Statement

The cross-domain performance degradation observed in this study highlights the potential risks of deploying models that appear accurate in controlled settings but fail on diverse real-world texts, which could lead to mismatched educational materials and negatively affect learners.

Acknowledgments

Generative AI tools were used to reduce the length of a much longer version of this article. Other than the insight in 5.4, all technical content, experimental design, results, and analyses are the authors' own work.

References

- Abdel-Karim Al-Tamimi, Manar Jaradat, Nuha Aljarrah, and Sahar Ghanim. 2014. AARI: Automatic Arabic readability index. *Information Technology Journal*, 13:1350–1357.
- Sultan Almujaivel, Damith Premasiri, Tharindu Ranasinghe, Mo El-Haj, and Ruslan Mitkov. 2025. [Complex concept-based readability estimation from Arabic curriculum](#). *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(11):1–21.
- Sarah Alzubi. 2026. [Robust and interpretable cross-domain arabic readability prediction using hybrid](#)

- modeling: Evidence on the limitations of transformer-based classifiers. Master's thesis, Brigham Young University, April.
- Mohammed Attia, Younes Samih, and Yo Ehara. 2023. [Statistical measures for readability assessment](#). In *Proceedings of the Joint 3rd International Conference on Natural Language Processing for Digital Humanities and 8th International Workshop on Computational Linguistics for Uralic Languages*, pages 153–161, Tokyo, Japan. Association for Computational Linguistics.
- Violetta Cavalli-Sforza, Hind Saddiki, and Naoual Nassiri. 2018. [Arabic readability research: Current state and future directions](#). *Procedia Computer Science*, 142:38–49.
- 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. ACM.
- Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.
- Michael A. Covington and Joe D. McFall. 2010. [Cutting the Gordian knot: The moving-average type–token ratio \(MATTR\)](#). *Journal of Quantitative Linguistics*, 17(2):94–100.
- Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. [Linguistic features for readability assessment](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17. Association for Computational Linguistics.
- Mahmoud El-Haj and Paul Rayson. 2016. OSMAN – a novel Arabic readability metric. *Proceedings of LREC 2016*.
- Mo El-Haj, Sultan Almujaivel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. DARES: Dataset for Arabic readability estimation of school materials. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. European Language Resources Association.
- Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025. [A large and balanced corpus for fine-grained Arabic readability assessment](#). *arXiv preprint*.
- Ahmed Elshabrawy, Muhammed AbuOdeh, Go Inoue, and Nizar Habash. 2023. [CamelParser2.0: A state-of-the-art dependency parser for Arabic](#). In *Proceedings of ArabicNLP 2023*, pages 170–180. Association for Computational Linguistics.
- Jonathan Neil Forsyth. 2014. Automatic readability detection for Modern Standard Arabic. Master's thesis, Brigham Young University.
- Joseph Marvin Imperial. 2021. [BERT embeddings for automatic readability assessment](#). In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 611–618. INCOMA Ltd.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104. Association for Computational Linguistics.
- Nouran Khallaf and Serge Sharoff. 2021. [Automatic difficulty classification of Arabic sentences](#). *arXiv preprint*.
- Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. [Pushing on text readability assessment: A transformer meets handcrafted linguistic features](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686. Association for Computational Linguistics.
- Ho Hung Lim, Tianyuan Cai, John S. Y. Lee, and Meichun Liu. 2024. Robustness of hybrid models in cross-domain readability assessment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. European Language Resources Association.
- 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.
- Tarek Naous, Michael J. Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. [ReadMe++: Benchmarking multilingual language models for multi-domain readability assessment](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Naoual Nassiri, Violetta Cavalli-Sforza, and Abdelhak Lakhouaja. 2019. [MoSAR: Modern Standard Arabic readability corpus for L1 learners](#). In *Proceedings of the 4th International Conference on Big Data and Internet of Things*, pages 1–7. ACM.
- Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMEL tools: An open source Python toolkit for Arabic natural language processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032. European Language Resources Association.
- Mohamed Amine Ouassil, Mohammed Jebbari, Rabia Rachidi, Mouaad Errami, Bouchaib Cherradi, and Abdelhadi Raihani. 2024. [Enhancing Arabic text readability assessment: A combined BERT and BiLSTM approach](#). In *2024 International Conference on Circuit, Systems and Communication (ICCSC)*, pages 1–7. IEEE.

- Sowmya Vajjala. 2022. Trends, limitations and open challenges in automatic readability assessment research. *arXiv preprint arXiv:2105.00973*.
- Frans Van Der Sluis and Egon L. Van Den Broek. 2025. [Model interpretability enhances domain generalization in the case of textual complexity modeling](#). *Patterns*, 6(2):101177.
- Ahmed Wasfy, Omer Nacar, Abdelakreem Elkhateb, Mahmoud Reda, Omar Elshehy, Adel Ammar, and Wadii Boulila. 2025. QARI-OCR: High-fidelity Arabic text recognition through multimodal large language model adaptation. *arXiv preprint arXiv:2506.02295*.

Predicting Item Difficulty and Generating Reading Comprehension Items via an Annotated Repository

Radhika Kapoor, Mayank Sharma, Sang T. Truong, Maria Araceli Ruiz-Primo,
Nick Joseph Haber, Benjamin W. Domingue

Stanford University, USA

{rkap786, masharma, sttruong, aruiz, nhaber, bdomingu}@stanford.edu

Abstract

Prediction of item difficulty from its text content is of substantial interest for automated generation of test items. In this paper, we focus on the related problem of recovering IRT-based difficulty when the data originally reported item p-value (percent correct responses). We model this item difficulty using a repository of reading passages and student data from US standardized tests from New York and Texas for grades 3-8 spanning the years 2018-23. This repository is annotated with meta-data on (1) linguistic features of the reading items, (2) test features of the passage, and (3) context features. Using a penalized regression model, we achieve an RMSE of 0.59 (compared to a 0.92 baseline) and a 0.77 correlation between true and predicted difficulty. We further evaluated the impact of LLM embeddings (ModernBERT, BERT, and LLaMA), finding that they marginally improve performance but function effectively as standalone alternatives to traditional linguistic features. Finally, we demonstrate how this difficulty prediction model powers a publicly available, human-in-the-loop tool for generating reading comprehension items.

1 Introduction

Education systems use reading comprehension tests to evaluate student achievement and growth, to monitor average school and teacher performance, and to diagnose students for learning disorders. Creating items for these tests can be an expensive and time-intensive endeavor (Case and Swanson, 1998; Rudner, 2009), especially for standardized tests in K-12 schools. Conventionally, subject matter experts (e.g., teachers or reading specialists) create the items used in standardized tests (Attali et al., 2014). Items are then piloted in schools and student response data is collected to estimate item properties like difficulty and to confirm that items are measuring student knowledge as intended (Lane

et al., 2016, Chapter 1).¹ However, items developed by subject matter experts might not perform as expected when taken to the field (Bejar, 2012; Reid, 1991; Impara and Plake, 1998; Dee and Domingue, 2021).

Education researchers have long been interested in modeling the difficulty of test items. Improved item difficulty models could support subject matter experts in creating items as well as reduce field testing time and costs (Bejar, 1991; Gorin and Embretson, 2006). However, this field has achieved mixed results, as modeling and predicting item difficulty is complex, and requires advanced text analysis and annotation to extract relevant item features (Gorin and Embretson, 2006).

Large Language Models (LLMs) have recently shown the potential to generate items with desirable characteristics like difficulty and to automate scoring of text-based responses (Ferrara et al., 2018; Huang et al., 2017; White et al., 2022; Attali et al., 2022; Zelikman et al., 2023) possibly reducing the time and cost associated with these tasks (Acharya et al., 2023). In this paper, we leverage LLMs with item and respondent features to predict mean difficulty of reading comprehension items. We build a repository of items using reading comprehension items from New York and Texas, administered to Grades 3-8 students spanning 5 years between 2018 and 2023. This repository contains the item text (reading passage, question, and distractors), and the state-level average percent correct responses (p-value). Items are annotated with (1) linguistic features of the reading passage, including their lexical, semantic, and syntactic characteristics and readability metrics (2) test features of the item, such as the presence of highlighted or bold text (3) respon-

¹New York: <https://www.nysed.gov/state-assessment/nysed-test-development-process>; Texas: <https://tea.texas.gov/texas-schools/accountability/academic-accountability/performance-reporting/8-w-hy-does-texas-field-test.pdf>

dent's state, grade, and year (4) LLM-generated text embeddings for reading comprehension items.

We focus on how effectively LLM-generated embeddings predict item difficulty in an attempt to understand whether transformer-based models can augment traditional psychometric methods used to generate items. Our models achieve an RMSE of 0.59 compared to a baseline of 0.92, and a correlation of 0.77 between true and predicted item difficulty. Previous results for predicting difficulty using item content achieved correlation of 0.3-0.7, and smaller RMSE improvements (Ha et al., 2019; Huang et al., 2017). We make the following contributions to literature: First, to our knowledge, this is the first publicly available work to examine automatic difficulty modeling for reading items for standardized tests in US schools. Second, we also convert mean p-value of items from different grades to a vertical logit scale using a Rasch IRT model, which could be used in future work to aggregate response data collected from different populations. Third, we compare difficulty prediction from different types of item features: linguistic features, test features, pre-trained LLM embeddings, and respondent context. We report similar results for prediction from linguistic features or LLM embeddings, suggesting that any of these classes of metrics could be used by themselves for prediction. Fourth, we compare the performance of three LLMs (BERT-base, ModernBERT, and LLaMA), as well as different prompts for generating LLM embeddings. We report similar results for these variations. Finally, we introduce a publicly available tool designed for practitioners and researchers to generate reading comprehension items with predicted difficulty. The tool can be found at <https://item-generator-production.up.railway.app/>.

2 Background

2.1 Features of Reading Comprehension Items

A reading comprehension item in a test is usually structured as a passage that the respondents read, followed by a question or prompt, and then by several options from which respondents select one response. The item might also include a few other components: directions about the passage or the item (e.g., "read the passage and answer the questions that follow"), images or figures (e.g., to highlight part of the story), and tables (e.g., with data

related to the story). This section describes the features of reading comprehension items that could predict their difficulty.

2.1.1 Features of Reading Passages

Reading standards such as the Common Core Standards for English Language Arts & Literacy² are tied to these complexity measures, both for teaching and tests. Text complexity of reading passages has been captured using (1) descriptive metrics, such as the count of words in passage, number of sentences, and number of paragraphs (2) linguistic complexity, which is influenced by factors such as complexity of words (measures include word length and word frequency), syntactic complexity (measures include modifier propositional density or count of adjectives per 1000 words), and text cohesion (defined as words and ideas overlapping throughout the text) (Graesser et al., 2014; McNamara et al., 2014).

Corpus analysis tools can parse text for their linguistic features. A popular corpus analysis tool is the Coh-Metrix, which offers over 100 measures of vocabulary, syntax, semantics, and other linguistic features. Coh-Metrix also reports aggregate measures of text cohesion and readability using metrics like Flesch-Kincaid, latent semantic analysis and principal components analysis (McNamara et al., 2014). Measures from corpus analysis tools, such as length of passage, concreteness or abstractness of text, and vocabulary level have been reported to be correlated with difficulty for reading questions (AlKhuyaey et al., 2023; Choi and Moon, 2020).

In addition to the reading complexity of the passage, its layout or structure might also affect student performance. The layout here refers to reading cues that point students to important aspects of the content (such as text segmentation), the inclusion of boxes or footnotes, directions that point readers to important vocabulary, helpful images, and options for text-to-speech in computer-based reading (Lawrence et al., 2022; Mize et al., 2020).

2.1.2 Features of Items

Item features affect their difficulty by changing their interpretability. Some features can make it easier to locate and retrieve information from text (Lawrence et al., 2022; Le Hebel et al., 2017; Ruiz-Primo and Li, 2015). For example, an item can include relevant sentences from the passage in a

²<https://achievethecore.org/page/2725/text-complexity>

text box or definitions of difficult words. Other features emphasize key aspects of items. For instance, the word “agree” in the item is emphasized: “Which of the following statements would *<name of character>* in the story **agree** with?”. In multiple choice questions, the distractors or response options also affect difficulty; for instance, some distractor options could be very similar to the correct answer. The type of content (e.g., use of images) might also influence student performance.

2.2 Item Difficulty Modeling

2.2.1 Automatic generation of items

Automated item generation was first proposed in the 1970s (Gierl and Haladyna, 2013), though the field saw a resurgence in the 2000s with the need for more item creation with computerized tests, as well as with an increase in computational power. A type of automatic item generation uses expert-created templates, item schemes or structures, parts of which can be populated using algorithms (Embretson and Yang, 2006; Kurdi et al., 2020). Such cognitive item model templates are specific to the item type being created.

With recent advances, LLMs can be prompted to generate items without additional tuning or information (“zero-shot” generation) or with some examples in context (“one-shot” or “few-shot” generation) (Attali et al., 2022; Kurdi et al., 2020). LLMs can also be prompted using cognitive models, templates, or rules to create items with desired properties (Gierl and Haladyna, 2013; Kurdi et al., 2020). These templates can specify the type of item to be generated (e.g., fill-in-the-blank questions, multiple-choice questions, open-ended questions), provide further instructions based on item type (e.g., the number of distractors and correct responses for multiple choice items), specify reasoning to be elicited (e.g., recall or inference), and even state the desired item difficulty (e.g., “create an easy item that 70% of respondents would get right”) (Sayin and Gierl, 2024). While LLMs can be used to generate big quantities of test items (Kurdi et al., 2020), their quality, especially properties like difficulty, is unknown, making them hard to use reliably in educational testing.

2.2.2 Item difficulty modeling

The problem of item generation is fundamentally linked with the problem of item difficulty modeling. Items are created with an implicit or explicit difficulty level, defined by human experts using

heuristics or by statistical modeling. Popular models used for statistical modeling include supervised learning approaches such as linear regression, decision trees, and random forests. Approaches like deep learning and neural nets have also been leveraged but are less common, likely because of the smaller size of education datasets and the desire for inference in addition to model prediction.

Statistical item difficulty modeling approaches usually have the following components: (1) A method to parse item text and images: this can use NLP and image parsers, linguistic corpus analysis tools, and human annotation (2) Model to predict difficulty. Recently, difficulty modeling has been supplemented by embeddings from LLMs (Kurdi et al., 2020; Sharpnack et al., 2024). Embeddings can be extracted from pre-trained models, or models could be fine-tuned. In the shared task organized during the BEA workshop at NAACL 2024 (Yaneva et al., 2024), item difficulty (defined as the percentage of respondents who responded incorrectly) was modeled for 667 MCQ items from the United States Medical Licensing Examination. The best prediction model achieved RMSE of 0.29 compared to a baseline RMSE of 0.31 from a dummy regressor model; the best model used a combination of LLM embeddings and other item features. In a similar exercise with 12,038 MCQ items from the United States Medical Licensing Examination, the best model used a combination of embeddings from Word2Vec and EIMo and linguistic features together, achieving RMSE of 22.45 compared to a baseline RMSE of 23.65.

The results of existing item difficulty prediction efforts have achieved limited success. A key constraint is that education settings rarely provide sufficiently large datasets for fine-tuning LLMs.

3 Data

The dataset contains multiple-choice items from the New York State Testing Program (NYSTP) and Texas STAAR reading comprehension standardized tests for Grades 3-8. NYSTP data is available for the years 2018, 2019, 2022, and 2023, and Texas STAAR data is available for 2019, 2021, and 2022. In total, there are 1076 items based on 170 passages across grades and years from both states (see item counts for grades by state and year in Appendix A).

Features available for passages and related multiple-choice items include item difficulty, defined as mean percent correct response at the state,

year, and grade level. Appendix B reports mean percent correct responses (i.e., the item's *p-value*³) across items by grade, state, and year. Note that mean accuracy for both states centers around 60% across grades and years. Student acquisition of additional reading comprehension skills is being offset by increases in overall item difficulty. As a consequence, reported difficulties or p-values cannot be directly compared across grades. In other words, let us say there is a Grade 3 item with p-value 0.60 (i.e., 60% of 3rd graders get this item right) and a Grade 8 item with p-value 0.60. The Grade 3 item is presumably easier than the Grade 8 item, since Grade 8 students are likely to be better readers; however, the p-values reported in the dataset would consider these items to be equally difficult. We convert these p-values to a common vertical logit scale to make them comparable for difficulty modeling. This is described in more detail in the Methods section.

4 Methods

This section describes how the dataset is processed for item difficulty prediction. Section 4.1 describes how mean p-values across grades and years are rescaled to a uniform logit vertical scale. Section 4.2 describes annotation of items with linguistic, test, and context features used as inputs in the prediction model. Section 4.3 describes how embeddings are generated from pre-trained LLMs. Finally, Section 4.4 then describes how these inputs are used to predict item difficulty using a penalized regression model.

4.1 Rescaling p-values

We use publicly available estimates of average student abilities across grades to convert average p-value to a common vertical scale. Appendix C shows the NWEA MAP 2020 reading scale⁴, which reports mean grade-level achievement norms for over 500,000 students attending public schools across 50 states. These average scores are used to transform the p-values from a specific grade to a scale that can then be used for direct comparison; for example, on our derived scale the 3rd grade item with a p-value of 0.6 will have a lower value than the 8th grade item with a p-value of 0.6 given the growth from 200.74 to 220.93 observed on the NWEA scale.

³Discussed in Chapter 14 in Crocker and Algina (1986)

⁴<https://www.nwea.org/uploads/2020/02/NY-MAP-Growth-Linking-Study-Report-2020-07-22.pdf>

4.1.1 Rescale p-values to IRT difficulty

To rescale the p-values to the logit scale, we take advantage of a basic feature of 1PL IRT models (von Davier, 2016) that suggests a relationship between such p-values and item-person characteristics:

$$\Pr(y = 1) = p_{ij} = \sigma(\theta_i + b_j) \quad (1)$$

where θ is the relevant person ability, b is the item easiness, and σ is a sigmoid function (the logistic sigmoid, $\sigma(x) = \frac{1}{1+\exp(-x)}$, which we also use here). Note that we predict item easiness rather than item difficulty, to remain consistent with the original dataset, which reports p-values as measures of easiness. We don't have access to the relevant θ_i and b values that generated the data here, but we utilize the fact that they, in particular average θ values by grade level, are publicly reported for similar groups. We thus do the following. We denote the mean ability for students in a given grade (using values in Appendix C) as $\bar{\theta}_g$. For each item j given to students in grade g , we can calculate easiness b_j based on the relationship between the p-value p_j and the above Equation 1. Specifically, we solve for b_j such that

$$p_{ij} = \frac{1}{1 + \exp(-(\bar{\theta}_g + b_j))} \quad (2)$$

$$\iff b_j = -\bar{\theta}_g + \log\left(\frac{p_j}{1 - p_j}\right) \quad (3)$$

We use the resulting b_j values in our subsequent modeling.

4.1.2 Adjustment Results

In the analysis below, $\bar{\theta}_g$ is drawn from the NWEA Spring 2020 reading vertical scale. Figure 1 shows the results of the adjustment process. The top panel shows grade-level mean b_j values for NY and Texas, mapped from p-values using $\bar{\theta}_g$ from NWEA 2020 reading achievement scores. The bottom panel shows the distribution of the unadjusted and adjusted p-values by grade. The unadjusted mean p-values (bottom left panel) are all distributed around 0.6 with no increasing pattern as grade levels increase. This pattern is adjusted on the logit scale in the bottom right panel. Note that adjusted Grade 3 items are easiest on average, while Grade 8 items are hardest. Hence, an average p-value of 0.6 for Grade 3 corresponds to 0.3 on the adjusted scale, and maps to -1.69 for Grade 8.

Note one critical fact. The values on the y-axis in Figure 1 are based on the NWEA Scale in Appendix C. While the NWEA scale is a high-quality

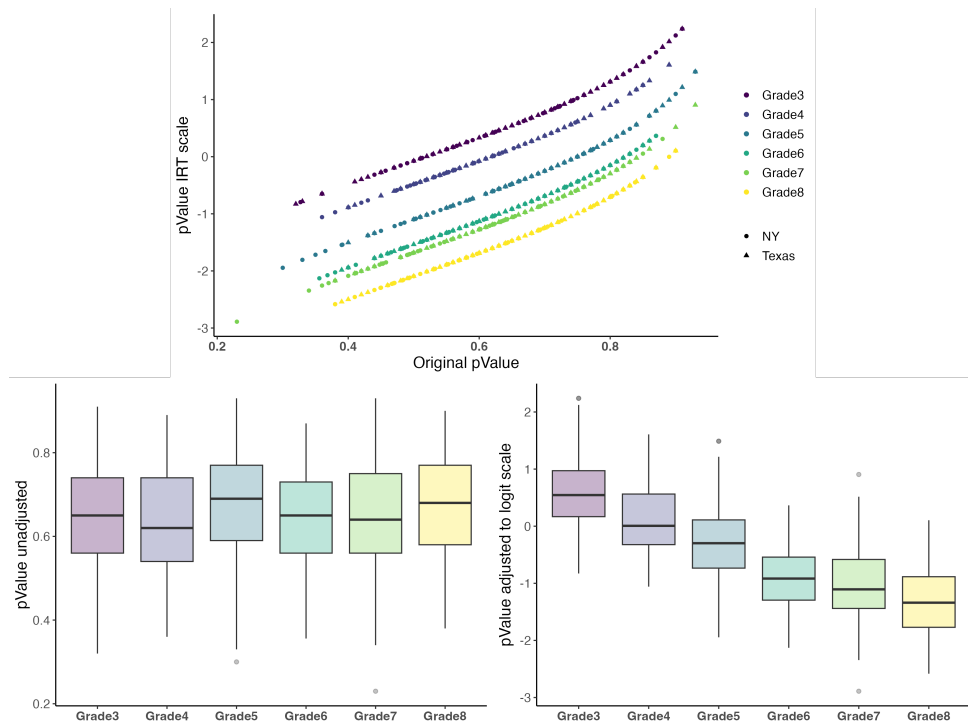


Figure 1: Conversion of p-value to IRT Easiness by Grades

Note. (Top): Conversion of p-value to IRT scale by Grade; (Bottom Left): Unadjusted p-values by Grades; (Bottom Right): Adjusted Mean Easiness by Grades on the Logit Scale

scale derived from large samples, it is one of several scales that could provide a common metric across grades (Dadey and Briggs, 2012). Given that this choice of $\bar{\theta}_g$ is somewhat arbitrary, we consider a range of potentially plausible $\bar{\theta}_g$ values in ancillary robustness analyses (discussed in Appendix G).

4.2 Annotation of item bank

Items are annotated with three categories of features: context, test characteristics, and linguistic text analysis measures.

4.2.1 Context Features

Respondent context characteristics are expected to be some of the most important predictors of difficulty. In this dataset, the context variables are the state, year, and grade of respondents, for which average difficulty values are reported.

4.2.2 Test Features

Test features refer to the presentation or design elements in a test which could influence response. For the passage, this includes the use of highlighted (bolded, italicized, or underlined) text to emphasize important elements. For questions, this refers to the use of highlighted text (e.g., the use of bold text

in “6. Which sentence **best** states the main idea of ‘Around the World’?”), or inclusion of relevant text or sentences from the reading passage. For example, in the following question, the sentence in bold is repeated from the passage: “Read this sentence from paragraph 6. **‘Traveling east across the Atlantic, Nellie took just one bag to move quickly.’** How is this detail important to paragraph 1?”. The inclusion of these test characteristics is captured as indicator variables (=1 when given test feature is present in the item). Appendix D lists the variables that capture relevant test characteristics about an item.

4.2.3 Text Analysis Features

Measures of linguistic complexity of reading passages are generated using Coh-Metrix software package and included as predictors for the analysis (Appendix E). The indicators are divided into the following categories: (1) Descriptive metrics, such as count of paragraphs, count of sentences, mean number of words in a sentence, and number of letters or syllables in a word (2) Readability metrics, such as Flesch-Kincaid score and Coh-Metrix L2 Readability (3) Ease of reading or text “easability”, estimated through principal components analysis

(PCA) components, measuring syntactic simplicity, vocabulary concreteness, referential cohesion, and deep cohesion (4) Text cohesion, where more cohesive texts are easier to read; measured by word overlap across text or semantic overlap in text (5) Deep cohesion in text, measured by use of causal verbs like *although* and *overall*, (6) Syntactic complexity, measured by use of more complex syntax such as density of noun phrases or syntax similarity of complex sentences (7) Word complexity, based on word frequency or psychological traits of words such as age of acquisition. These measures are generated by the Coh-Metrix software (McNamara et al., 2014), except for the Flesch-Kincaid score (generated by readability package in Python⁵), and count of paragraphs in passage and word count of passage, which was generated in R.

4.3 Sentence Embedding from Large Language Models

LLM-generated embeddings for question text and distractors are used as predictors in difficulty modeling. Distractors are tagged with correct answer or wrong answer as appropriate. The question text, correct answer and distractors, and passage are merged into one input string, for which embeddings are generated (Appendix F). We experiment with BERT (Devlin et al., 2018), ModernBERT base model with 149M parameters (Warner et al., 2024) and LLaMA-3.1-8B (Dubey et al., 2024) for the generation of sentence embeddings.

To generate embeddings from ModernBERT⁶ and BERT⁷, each item is converted to a statement by merging the question text and available options as follows: (1) question text (2) a tag for correct answer, followed by the correct option (3) a tag for wrong answer, followed by the first wrong answer in the question. This is repeated for the remaining wrong answers (4) Reading passages. The objective is to capture interactions between the passage, question text, and distractors. The BERT tokenizer is used to convert these sentences into token IDs. The embeddings are derived from the last hidden state of the model. Specifically, for each sentence, we determine the position of the last meaningful token by identifying the index of the final non-padding token. The hidden state of the

final token is extracted. Each sentence produces a tensor whose size depends on the number of tokens times the embedding dimension. Because sentence lengths vary, these tensors are averaged across tokens to produce fixed-size embeddings (768 for BERT and ModernBERT, 4096 for LLaMA). This tensor is the final embedding output for the item. The dimensionality of these embeddings is also compressed using PCA to capture 80% of variation. PCA on embeddings also makes it harder to recover the question text and options from the embeddings, which would be a test security concern for high-stakes tests.

We also experimented with other methods of generating embeddings from BERT by removing the reading passage from the text input to tokenizer. We also generate cosine similarity between correct answer and wrong answers as new predictor variables. The results have some sensitivity to the method of embedding generation, but main conclusions do not change (Appendix G, Table 9).

4.4 Prediction Model

Item-level difficulty (i.e., b_j from Eqn 3) is predicted through a Ridge (L_2) penalized regression that tuned the regularization penalty parameter (λ)⁸ via 5-fold cross-validation (repeated and averaged 5 times with a 80% train, 20% test split). All predictor variables are centered at 0 with standard deviation set to 1. The performance of the prediction models is measured by two indicators: RMSE, and Pearson Correlation Coefficient between the true (y_j) and predicted outcome ($\hat{y}_j$) variable (AlKhuyaey et al., 2023). RMSE is sensitive to the training dataset, and hence is meaningful only for within-dataset model comparisons. The correlation coefficient, however, is better suited for benchmarking this model against results reported in the literature.

5 Results

The predictors include the following categories of variables: (1) context characteristics of state, grade, and year (2) test characteristics (3) linguistic features of reading passage and item (4) LLM-generated embeddings (either the full set of embeddings or PCA components that capture 80% of the variation). The baseline model for comparison sets the predicted difficulty to the mean difficulty for

⁵<https://pypi.org/project/readability/>

⁶<https://huggingface.co/answerdotai/ModernBERT-base>

⁷<https://huggingface.co/google-bert/bert-base-cased>

⁸Elastic net models that tune the ratio between ridge and lasso (α) always converged to the more parsimonious ridge model

Table 1: Results for Predicting Item Difficulty

	RMSE		Correlation	
	Train	Test	Train	Test
Results from human annotated features				
State, Grade, Year	0.60	0.64	0.74	0.72
Test features	0.89	0.92	0.13	0.05
Text analysis features	0.59	0.62	0.75	0.75
All annotated features and context	0.53	0.59	0.81	0.76
Results from LLM embeddings only				
BERT embeddings	0.60	0.66	0.76	0.71
LLaMA embeddings	0.44	0.66	0.89	0.70
ModernBERT embeddings	0.58	0.62	0.77	0.76
Results from LLM embeddings and annotated features				
Annotated features & BERT embeddings	0.60	0.64	0.76	0.73
Annotated features & PCA on BERT embeddings	0.58	0.64	0.76	0.72
Annotated features & LLaMA embeddings	0.47	0.62	0.87	0.75
Annotated features & PCA on LLaMA embeddings	0.59	0.63	0.75	0.73
Annotated features & ModernBERT embeddings	0.57	0.61	0.78	0.76
Annotated features & PCA on ModernBERT embeddings	0.59	0.63	0.75	0.73
Results from LLM embeddings, annotated features, and context				
All features & BERT embeddings	0.59	0.64	0.77	0.74
All features & PCA on BERT embeddings	0.52	0.60	0.81	0.76
All features & LLaMA embeddings	0.47	0.61	0.87	0.76
All features & PCA on LLaMA embeddings	0.52	0.59	0.81	0.76
All features & ModernBERT embeddings	0.57	0.61	0.78	0.77
All features & PCA on ModernBERT embeddings	0.52	0.59	0.81	0.77

Note. Outcome variable is easiness drawn from IRT scale based on NWEA MAP Reading scale Spring 2020

the item; this baseline RMSE was 0.92. This serves as a comparison point for model performance as measured by RMSE, which are dependent on the sample data set.

5.1 Prediction results from annotated features and embeddings

The best performing model uses all available features and PCA on embeddings from ModernBERT or LLaMA (Table 1). The best RMSE on test data is 0.59 and correlation between true and predicted easiness is 0.77.⁹ It is not surprising that the best model uses all available features for prediction. We also note that PCA on embeddings performs marginally better than actual embeddings; this is likely because of the small dataset size compared to the number of predictors.

⁹Appendix G reports the results when other vertical scales are used to convert p-values to IRT easiness: for different scales, RMSE varies between 0.53 and 0.61, and correlation varies between 0.75 and 0.85. The RMSE is 3.24 and the correlation is 0.96 when different scales are used for pre- and post-2020 years. This is not strictly comparable with other results which use the same scales for all years

5.2 Prediction results from LLM embeddings only

Results from only LLM embeddings, without using any respondent context characteristics or linguistic analysis of text, are close to the best model (Table 1). The correlation of true and predicted difficulty is 0.76 for ModernBERT with an RMSE of 0.62. When all additional features are added, model performance increases marginally for RMSE to 0.61 and correlation to 0.77. BERT and LLaMA show slightly bigger improvements in model performance when all features are added: RMSE decreases from 0.66 when only BERT embeddings are included to 0.64 when all features are included; the correlation between true and predicted difficulty in the test dataset increases from 0.71 to 0.74. For LLaMA, RMSE decreases from 0.66 to 0.61, and correlation increases from 0.70 to 0.76.

5.3 Prediction results from annotated features

Model performance for annotated features (context, test, and text analysis) is also close to the best model. This is largely because of the text analysis features, which are able to predict item difficulty, suggesting that LLMs and text analysis features may be capturing similar information. Text

analysis features offer the added benefit of being interpretable by humans and require less computation power, which may be an advantage over using LLMs.

Examining the standardized ridge regression coefficients reveals which features are most important for prediction (Appendix F.2, Table 7). Among text analysis features, the strongest predictors are measures of referential cohesion and syntactic complexity. Specifically, noun phrase overlap between adjacent sentences (CRFANP1) and noun phrase density (DRNP) are the two largest predictors, both negatively associated with item easiness — passages with denser and less cohesive noun phrase structure tend to produce harder items. Among word-level features, higher word frequency (WRD-FRQa) and greater imageability of content words (WRDIMGc) are both associated with easier items. Notably, traditional readability metrics such as Flesch-Kincaid rank poorly relative to these deeper linguistic features, suggesting that surface-level readability scores commonly used in educational practice may be insufficient for capturing item difficulty.

The overall prediction power of test features by themselves is low, as would be expected given there are only four such variables. However, one test feature appears among the top predictors overall: the presence of highlighted text in the question (*ques_text_highlight_yn*) is positively associated with item easiness, consistent with prior work suggesting that question-level presentation cues help students locate relevant information in the passage.

It is also important to note that these features can predict difficulty without including context characteristics (state, grade, year). These models could hence be used to predict average item difficulty before extensive field pilots, potentially reducing costs.

5.4 Automatic generation of items

The item difficulty model powers the Automatic Item Generator (<https://item-generator-production.up.railway.app/>), a web-based, human-in-the-loop tool that leverages the Claude API for item creation, along with the option for manual edits by users. This system facilitates the creation of assessment materials by (1) Item Generation: Reading items are generated by the Claude generative AI agent, using the existing item bank as context for generation. (2) Difficulty Calibration: Item difficulty scores are calculated on a logit scale

using ModernBERT embeddings.

The tool allows for specific parameter adjustments, including the target grade level, passage length, and the number of distractors. It also supports custom context, enabling researchers to integrate their own item generation manuals. Without additional prompting, the tool generates a synthetic dataset that mirrors the characteristics of standardized test items from New York and Texas.

6 Discussion

This paper aims to predict item difficulty using its text content. This approach bridges the gap between manual item creation and modern computational methods, providing insights into the nuanced interplay between text complexity and student response behaviors. The study underscores the transformative potential of transformer-based models, which can generate meaningful representations of items without fine-tuning or additional context information. By rescaling difficulties using IRT and vertical ability scales, the study addressed the challenge of comparing average difficulty across grades, offering a scalable solution when there is no common population or anchor items for traditional test equating and linking approaches. This item difficulty prediction model could be useful for stakeholders like teachers or testing firms, who would be interested in creating items with desirable item properties like difficulty. For instance, an item generated through generative AI could be assigned a difficulty score using this model; this could then be used to assemble appropriate tests.

Our model demonstrated strong performance in estimating item difficulty, achieving a correlation coefficient of 0.77 between predicted and observed difficulty - a result obtained both by ModernBERT embeddings alone and by models combining all available features with LLM embeddings. To the best of our knowledge, this correlation between true and predicted difficulty is one of the best for multiple-choice reading comprehension test items. The RMSE is high, but we expect that it is a function of the constraints of the dataset; the dataset we use is relatively small and the outcome variable is average item difficulty converted to a logit scale. The best model also reduces RMSE from a baseline value of 0.92 to 0.59. Other work that predicts difficulty as a continuous measure reports Pearson's correlation between 0.38 and 0.4 (Huang et al., 2017) and a reduction in RMSE from 23.65

to 22.45 (Ha et al., 2019).

The best results were obtained for the models that combined text analysis and human generated features with LLM embeddings. Surprisingly, in our dataset, using LLM-based embeddings or text analysis features was close to the best performing model. These findings suggest that LLMs could predict item difficulty without additional linguistic analysis or human annotations. Embeddings were generated from three models, BERT, ModernBERT and LLaMA; the results are also similar for the three LLM models. LLaMA embeddings were computationally more expensive to generate but their model performance was better. ModernBERT achieved the best performance, suggesting it might be best suited to such an item difficulty modeling task. BERT's strong performance despite being the simplest model could be a function of the limited size of the dataset, where variations in embedding generation might not lead to improvements in results.

Limitations

While the findings affirm the utility of automated methods, they also highlight several limitations of static embeddings in fully capturing dynamic student interactions. First, the reliance on state-specific datasets from New York and Texas may limit the generalizability of the findings to other educational contexts or student populations. Second, the embeddings generated from LLMs are dependent on pre-trained models that may not be optimally tuned to K-12 educational texts or standardized test items, potentially missing domain-specific nuances. Lastly, the model's predictions are based on aggregated item difficulty, which may oversimplify individual differences in student abilities or test-taking contexts.

Future research should focus on expanding the dataset to include diverse state tests and international benchmarks, such as PISA or data high-stakes tests, to enhance the robustness and applicability of the model. Additionally, fine-tuning LLMs on domain-specific corpora could further improve their ability to capture educational nuances. A comprehensive evaluation of item difficulty prediction should explore its impact on downstream tasks, such as automated item generation and field testing, to validate the practical utility of the proposed model in educational settings. Last, but perhaps most important, this model could be used by teach-

ers or test creators to get early feedback on item properties.

Ethical Considerations

The use of automated tools in educational assessment raises important ethical considerations. Item difficulty prediction models, including the one presented here, should be understood as decision-support tools rather than replacements for human judgment. Decisions about item inclusion, exclusion, or revision in high-stakes assessments should remain with qualified subject matter experts, psychometricians, and assessment professionals who can evaluate item quality in context.

The automatic item generation tool described in this paper is intended to assist practitioners in creating reading comprehension items more efficiently. However, items generated by any AI system require careful human review before use in consequential testing contexts. Automated generation does not guarantee that items are free from bias, culturally appropriate, or aligned with specific curriculum standards. Practitioners using such tools should be aware that generated items may reflect patterns and assumptions present in the training data, which in this case is drawn from two specific US state assessments. There is also a risk that over-reliance on difficulty prediction models could narrow the range of items developed, if practitioners consistently target items predicted to fall within a specific difficulty range.

References

- Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. LLM based generation of item-description for recommendation system. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1204–1207.
- Samah AlKhazaey, 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, Andrew Runge, Geoffrey T LaFlair, Kevin Yancey, Sarah Goodwin, Yena Park, and Alina A Von Davier. 2022. The interactive reading task: Transformer-based automatic item generation. *Frontiers in Artificial Intelligence*, 5:903077.
- 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.

- Isaac I Bejar. 1991. A generative approach to psychological and educational measurement. *ETS Research Report Series*, 1991(1):i–54.
- Isaac I Bejar. 2012. Item generation: Implications for a validity argument. In *Automatic item generation*, pages 40–55. Routledge.
- Susan M Case and David B Swanson. 1998. *Constructing written test questions for the basic and clinical sciences*. National Board of Medical Examiners Philadelphia.
- 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.
- Linda Crocker and James Algina. 1986. *Introduction to classical and modern test theory*. ERIC.
- Nathan Dadey and Derek C Briggs. 2012. A meta-analysis of growth trends from vertically scaled assessments. *Practical Assessment, Research & Evaluation*, 17(14):n14.
- Thomas S Dee and Benjamin W Domingue. 2021. Assessing the impact of a test question: Evidence from the “underground railroad” controversy. *Educational Measurement: Issues and Practice*, 40(2):81–88.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Susan Embretson and Xiangdong Yang. 2006. 23 automatic item generation and cognitive psychology. *Handbook of statistics*, 26:747–768.
- S Ferrara, J Steedle, and R Frantz. 2018. Item design with test score interpretation in mind. In *Item difficulty modeling: Lessons learned and future directions. Coordinated session at the annual meeting of the National Council on Measurement in Education*, New York, NY.
- Mark J Gierl and Thomas M Haladyna. 2013. *Automatic item generation: Theory and practice*. Routledge.
- Joanna S Gorin and Susan E Embretson. 2006. Item difficulty modeling of paragraph comprehension items. *Applied Psychological Measurement*, 30(5):394–411.
- Arthur C Graesser, Danielle S McNamara, Zhiqiang Cai, Mark Conley, Haiying Li, and James Pennebaker. 2014. Coh-matrix measures text characteristics at multiple levels of language and discourse. *The Elementary School Journal*, 115(2):210–229.
- 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.
- 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.
- 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.
- 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.
- Suzanne Lane, Mark R Raymond, Thomas M Haladyna, and 1 others. 2016. *Handbook of test development*, volume 2. Routledge New York, NY.
- Joshua F Lawrence, Rebecca Knoph, Autumn McIlraith, Paulina A Kulesz, and David J Francis. 2022. Reading comprehension and academic vocabulary: Exploring relations of item features and reading proficiency. *Reading Research Quarterly*, 57(2):669–690.
- Florence Le Hebel, Pascale Montpied, Andrée Tiberghien, and Valérie Fontanieu. 2017. Sources of difficulty in assessment: Example of pisa science items. *International Journal of Science Education*, 39(4):468–487.
- 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.
- Minnie Mize, Yujeong Park, Megan Schramm-Possinger, and Mari Beth Coleman. 2020. Developing a rubric for evaluating reading applications for learners with reading difficulties. *Intervention in School and Clinic*, 55(3):145–153.
- Jerry B Reid. 1991. Training judges to generate standard-setting data. *Educational Measurement: Issues and Practice*, 10(2):11–14.
- Lawrence M Rudner. 2009. Implementing the graduate management admission test computerized adaptive test. In *Elements of adaptive testing*, pages 151–165. Springer.
- Maria Araceli Ruiz-Primo and Min Li. 2015. The relationship between item context characteristics and student performance: The case of the 2006 and 2009 pisa science items. *Teachers College Record*, 117(1):1–36.

- Ayfer Sayin and Mark Gierl. 2024. Using openai gpt to generate reading comprehension items. *Educational Measurement: Issues and Practice*, 43(1):5–18.
- James Sharpnack, Kevin Hao, Phoebe Mulcaire, Clinton Bicknell, Geoff LaFlair, Kevin Yancey, and Alina A von Davier. 2024. Banditcat and autoirt: Machine learning approaches to computerized adaptive testing and item calibration. *arXiv preprint arXiv:2410.21033*.
- Matthias von Davier. 2016. Rasch model. In *Handbook of item response theory*, pages 31–48. Chapman and Hall/CRC.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). *Preprint*, arXiv:2412.13663.
- Julia White, Amy Burkhardt, Jason Yeatman, and Noah Goodman. 2022. Automated generation of sentence reading fluency test items. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.
- Victoria Yaneva, Kai North, Peter Baldwin, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, Brian Clauser, and 1 others. 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)*, pages 470–482.
- Eric Zelikman, Wanjing Anya Ma, Jasmine E Tran, Diyi Yang, Jason D Yeatman, and Nick Haber. 2023. Generating and evaluating tests for k-12 students with language model simulations: A case study on sentence reading efficiency. *arXiv preprint arXiv:2310.06837*.

A Item Counts by State, Grade, and Year

Table 2: Count of Items for Grades by State and Year

State	Grade	Item counts by year					Total
		2018	2019	2021	2022	2023	
NY	Grade 3	12	12		12	17	53
NY	Grade 4	12	12		12	17	53
NY	Grade 5	21	21		21	19	82
NY	Grade 6	21	21		21	19	82
NY	Grade 7	21	21		21	26	89
NY	Grade 8	21	21		21	26	89
Texas	Grade 3		34	28	34		96
Texas	Grade 4		32	32	32		96
Texas	Grade 5		29	34	34		97
Texas	Grade 6		36	36	36		108
Texas	Grade 7		37	37	38		112
Texas	Grade 8		40	39	40		119
Total		108	316	206	322	124	1076

B Mean Percent Correct Response (p-values) by State, Grade, and Year

Table 3: Mean Percent Correct Response (p-values) by State, Grade, and Year

State	Grade	Mean percent correct by year				
		2018	2019	2021	2022	2023
NY	Grade 3	0.60	0.65		0.65	0.60
NY	Grade 4	0.58	0.57		0.64	0.56
NY	Grade 5	0.62	0.64		0.65	0.60
NY	Grade 6	0.67	0.65		0.67	0.58
NY	Grade 7	0.55	0.60		0.66	0.59
NY	Grade 8	0.70	0.62		0.64	0.61
Texas	Grade 3		0.68	0.62	0.68	
Texas	Grade 4		0.66	0.62	0.70	
Texas	Grade 5		0.70	0.68	0.74	
Texas	Grade 6		0.64	0.64	0.65	
Texas	Grade 7		0.67	0.65	0.70	
Texas	Grade 8		0.70	0.67	0.71	

C NWEA Spring 2020 Reading Student Achievement Norms

Table 4: NWEA Spring 2020 Reading Student Achievement Norms by Grade

Grade	Score
Grade 3	200.74
Grade 4	204.83
Grade 5	210.19
Grade 6	215.36
Grade 7	216.81
Grade 8	220.93

D Passage and Item Characteristics Variables

Table 5: Variables: Passage and Item Characteristics

Variable	Description
item_order	Number indicating the order in which item appears after a passage
pass_highlight_yn	Indicator (1/0) if the passage text is highlighted
ques_text_ref_yn	Indicator (1/0) if the question text includes relevant paragraphs or sentences from text
ques_text_highlight_yn	Indicator (1/0) if the question text includes text in bold or underlined

E Variables Used as Predictors

E.1 Linguistic Features

Table 6: Table of Measures and Descriptions

Measure	Description
Descriptive metrics	
PassageWordCount	Word count of the passage
DESPC	Paragraph count, number of paragraphs
DESSC	Sentence count, number of sentences
DESPL	Paragraph length, number of sentences in a paragraph, mean
DESPLd	Paragraph length, number of sentences in a paragraph, standard deviation
DESSL	Sentence length, number of words, mean
DESSLd	Sentence length, number of words, standard deviation
DESWLsy	Word length, number of syllables, mean
DESWLsyd	Word length, number of syllables, standard deviation
DESWLlt	Word length, number of letters, mean
DESWLltd	Word length, number of letters, standard deviation
Readability metrics	
FK	Flesch-Kincaid score
RDFRE	Flesch-Kincaid Reading Ease
RDFKGL	Flesch-Kincaid Grade level
RDL2	Coh-Metrix L2 Readability
Text reading “easability”: Principal Components analysis mapped to dimensions of reading ease i.e., syntactic simplicity, word concreteness, referential cohesion, deep cohesion, verb cohesion, connectivity, and temporality	
PCNARz	Text Easability PC Narrativity, z score (e.g., tells a story)
PCNARp	Text Easability PC Narrativity, percentile
PCSYNz	Text Easability PC Syntactic simplicity, z score (e.g., sentences contain fewer words and simpler sentences)
PCSYNp	Text Easability PC Syntactic simplicity, percentile
PCCNCz	Text Easability PC Word concreteness, z score (e.g., more concrete words)
PCCNCp	Text Easability PC Word concreteness, percentile
PCREFz	Text Easability PC Referential cohesion, z score (e.g., words or ideas repeat across text)
PCREFp	Text Easability PC Referential cohesion, percentile
PCDCz	Text Easability PC Deep cohesion, z score (e.g., text contains causal and intentional connectives that aid with comprehension)
PCDCp	Text Easability PC Deep cohesion, percentile
PCVERBz	Text Easability PC Verb cohesion, z score (e.g., overlapping verbs in text)
PCVERBp	Text Easability PC Verb cohesion, percentile
PCCONNz	Text Easability PC Connectivity, z score (e.g., contains explicit adversative, additive, and comparative connectives)
PCCONNp	Text Easability PC Connectivity, percentile
PCTEMPz	Text Easability PC Temporality, z score (e.g., text has cues about temporality such as tense and aspect)
PCTEMPp	Text Easability PC Temporality, percentile
Referential cohesion: Word overlap across text, measured by indices of word and sentence overlaps	
CRFNO1	Noun overlap, adjacent sentences, binary, mean
CRFAO1	Argument overlap, adjacent sentences, binary, mean
CRFSO1	Stem overlap, adjacent sentences, binary, mean

Measure	Description
CRFNOa	Noun overlap, all sentences, binary, mean
CRFAOa	Argument overlap, all sentences, binary, mean
CRFSOa	Stem overlap, all sentences, binary, mean
CRFCWO1	Content word overlap, adjacent sentences, proportional, mean
CRFCWO1d	Content word overlap, adjacent sentences, proportional, standard deviation
CRFCWOa	Content word overlap, all sentences, proportional, mean
CRFCWOad	Content word overlap, all sentences, proportional, standard deviation
Referential cohesion: Semantic overlap in text measured by Latent Semantic Analysis	
LSASS1	LSA overlap, adjacent sentences, mean
LSASS1d	LSA overlap, adjacent sentences, standard deviation
LSASSp	LSA overlap, all sentences in paragraph, mean
LSASSpd	LSA overlap, all sentences in paragraph, standard deviation
LSAPP1	LSA overlap, adjacent paragraphs, mean
LSAPP1d	LSA overlap, adjacent paragraphs, standard deviation
LSAGN	LSA given/new, sentences, mean
LSAGNd	LSA given/new, sentences, standard deviation
Lexical diversity: Variety of words in a given text. Lexical diversity is lower and cohesion is higher when words are repeated across text	
LDTTRc	Lexical diversity, type-token ratio, content word lemmas
LDTTRa	Lexical diversity, type-token ratio, all words
LDMTLD	Lexical diversity, MTLTLD, all words
LDVOCD	Lexical diversity, VOCD, all words
Connectives: Incidence of connectives per 1000 words, which reflects cohesive links between ideas and text organization. Cohesion is higher when there is greater incidence of connectives.	
CNCAI	All connectives incidence
CNCCaus	Causal connectives incidence (e.g., <i>because, so</i>)
CNCLogic	Logical connectives incidence (e.g., <i>and, or</i>)
CNCADC	Adversative and contrastive connectives incidence (e.g., <i>although, whereas</i>)
CNCTemp	Temporal connectives incidence (e.g., <i>first, until</i>)
CNCTempx	Expanded temporal connectives incidence
CNCAdd	Additive connectives incidence (e.g., <i>and, moreover</i>)
CNCPos	Positive connectives incidence (e.g., <i>also, moreover</i>)
CNCNeg	Negative connectives incidence
Situation model: Representations of deeper meaning in text using Lexical Semantic Analysis and WordNet	
SMCAUSv	Causal verb incidence (e.g., <i>although, whereas</i>)
SMCAUSvp	Causal verbs and causal particles incidence
SMINTEp	Intentional verbs incidence
SMCAUSr	Ratio of causal particles to causal verbs
SMINTER	Ratio of intentional particles to intentional verbs
SMCAUSlsa	LSA verb overlap
SMCAUSwn	WordNet verb overlap
SMTEMP	Temporal cohesion, tense and aspect repetition, mean
Syntactic complexity: Includes measures of complex syntax in text and Syntactic Pattern Density; where higher density reflects more informational dense text with more complex syntax	
SYNLE	Left embeddedness, words before main verb, mean
SYNNP	Number of modifiers per noun phrase, mean
SYNMEDpos	Minimal Edit Distance, part of speech
SYNMEDwrd	Minimal Edit Distance, all words
SYNMEDlem	Minimal Edit Distance, lemmas

Measure	Description
SYNSTRUTa	Sentence syntax similarity, adjacent sentences, mean
SYNSTRUTt	Sentence syntax similarity, all combinations, across paragraphs, mean
DRNP	Noun phrase density, incidence
DRVp	Verb phrase density, incidence
DRAP	Adverbial phrase density, incidence
DRPP	Preposition phrase density, incidence
DRPVAL	Agentless passive voice density, incidence
DRNEG	Negation density, incidence
DRGERUND	Gerund density, incidence
DRINF	Infinitive density, incidence
Word complexity: information about parts of speech (e.g., relative frequency of types of word categories such as nouns, verbs and adverbs), word frequency (e.g., frequency of words used in text in CELEX database), and word complexity based on psychological ratings (e.g., age of acquisition of words, word concreteness)	
WRDNOUN	Noun incidence
WRDVERB	Verb incidence
WRDADJ	Adjective incidence
WRDADV	Adverb incidence
WRDPRO	Pronoun incidence
WRDPRP1s	First person singular pronoun incidence
WRDPRP1p	First person plural pronoun incidence
WRDPRP2	Second person pronoun incidence
WRDPRP3s	Third person singular pronoun incidence
WRDPRP3p	Third person plural pronoun incidence
WRDFRQc	Word frequency for content words using CELEX database, mean
WRDFRQa	CELEX Log frequency for all words, mean
WRDFRQmc	CELEX Log minimum frequency for content words, mean
WRDAOAc	Age of acquisition for content words, mean
WRDFAMc	Familiarity for content words, mean
WRDCNCc	Concreteness for content words, mean
WRDIMGc	Imagability for content words, mean
WRDMEAc	Meaningfulness, Colorado norms, content words, mean
WRDPOLc	Polysemy for content words, mean
WRDHYPn	Hypernymy for nouns, mean
WRDHYPv	Hypernymy for verbs, mean
WRDHYPnv	Hypernymy for nouns and verbs, mean

F Predictors for Item Difficulty Model

F.1 Visual representation of predictors in Item Difficulty Model

The figure shows a sample item from New York 2019 Grade 4 test. The box “**Input text for generating embeddings**” demonstrates how the passage and item is converted into a sentence. Embeddings are generated for this sentence. The boxes on the right show samples of annotated features for the item: **Linguistic features** shows some examples of text analysis of the reading passage. **Test features** show the relevant item design elements. **Context features** capture the item’s year, state, and targeted grade.

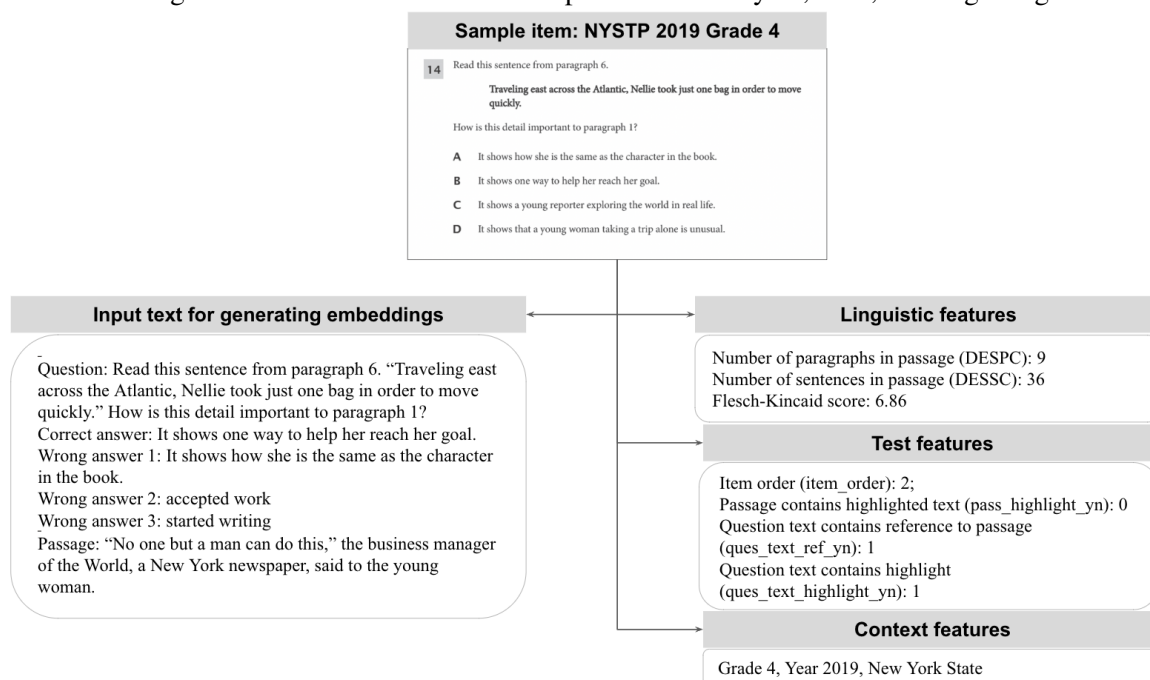


Figure 2: Predictors for Item Difficulty Model: Input Text to Generate Embeddings, and Linguistic, Test, and Context Features

F.2 Top predictors of Item Difficulty from Ridge Regression

Table 7: Top 20 Predictors of Item Difficulty from Ridge Regression (Text Analysis and Human-Annotated Features)

Rank	Variable	Coefficient	Absolute Value
Referential Cohesion			
1	CRFANP1 (Noun phrase overlap, adjacent sentences)	−0.127	0.127
14	LSAPP1 (LSA overlap, adjacent paragraphs)	−0.065	0.065
15	LSAGNd (LSA given/new, sentences SD)	+0.065	0.065
16	LSASS1 (LSA overlap, adjacent sentences)	+0.063	0.063
Syntactic Complexity			
2	DRNP (Noun phrase density)	−0.123	0.123
5	SYNNP (Modifiers per noun phrase)	−0.106	0.106
7	SYNMEDlem (Minimal edit distance, lemmas)	+0.104	0.104
19	DRGERUND (Gerund density)	+0.056	0.056
Descriptive Metrics			
3	DESSC (Sentence count)	+0.109	0.109
9	DESPC (Paragraph count)	−0.093	0.093
12	DESWC (Passage word count)	−0.069	0.069
20	PassageWordCount.gen (Passage word count, generated)	−0.056	0.056
Test Features (Human Annotated)			
4	ques_text_highlight_yn (Question text contains highlight)	+0.108	0.108
Connectives			
6	CNCNeg (Negative connectives incidence)	+0.104	0.104
10	CNCADC (Adversative & contrastive connectives)	−0.077	0.077
Word Complexity			
8	WRDFRQa (Word frequency, all words)	+0.097	0.097
11	WRDIMGc (Imageability, content words)	+0.070	0.070
13	WRDNOUN (Noun incidence)	+0.066	0.066
Situation Model			
17	SMCAUSwn (WordNet verb overlap)	+0.063	0.063
18	SMINTEp (Intentional verb incidence)	+0.061	0.061

Note. Coefficients are from a Ridge (L2) regression predicting IRT-scaled item easiness. All predictors are standardized (mean = 0, SD = 1). Positive coefficients indicate association with easier items; negative coefficients indicate association with harder items. Only text analysis and human-annotated features are shown; embedding predictors are excluded as their coefficients are not directly interpretable.

G Robustness Checks

G.1 Robustness to Choice of Grade-Level Growth Scale

This section examines the robustness of results to selecting an alternate vertical scale to convert p-values to IRT difficulty. Table 8 describes alternate scales that could be used: NWEA 2015 MAP reading achievement standard (reported separately for literary texts and informational text), NWEA 2020 reading achievement as measured in Fall or Winter, Texas STAAR 2023-24 performance standards (“approaches grade-level”, “meets grade level”, and “masters grade level” performance), and Texas STAAR 2017-18 readiness standard. These different scales were selected to account for differences in state contexts, as well as differences in grade level scales before and after 2020.

Results for test RMSE, as well as correlation between true and predicted RMSE are shown in Figure 3. We note that the results presented in the main section are conservative and do not fluctuate drastically with choice of scale. The greatest variation is observed when we use NWEA Informational Text 2015 scale for 2018 and 2019, and the NWEA 2020 Spring scale for 2021 to 2023. The RMSE increases to 3.24, likely due to the introduction of variation across years. However, correlation between true and predicted difficulty also increases to 0.96. Since it is difficult to obtain precise estimates of the change in vertical grade level scales before and after 2020, these results are not highlighted in the main paper.

Table 8: Alternate Vertical Growth Scales: NWEA MAP 2015, NWEA MAP 2020 and Texas Meets Grade Level Performance 2023-24

Grades	NWEA 2015		NWEA 2020			Texas STAAR 2023-24			Texas 2017-18
	Literary Text	Informational Text	Fall	Winter	Spring	Approaches grade-level	Meets grade-level	Masters grade-level	Readiness standard
Grade 3	192.4	191.6	186.62	195.91	200.74	1345	1467	1596	1386
Grade 4	201.2	200.7	196.67	202.5	204.83	1414	1552	1663	1473
Grade 5	207.9	207.4	204.48	210.19	210.98	1471	1592	1700	1508
Grade 6	212.3	212.1	210.17	213.81	215.36	1535	1634	1749	1554
Grade 7	216.3	216.1	214.2	217.09	216.81	1564	1669	1771	1603
Grade 8	220.0	220.0	218.9	220.52	220.93	1592	1698	1803	1625

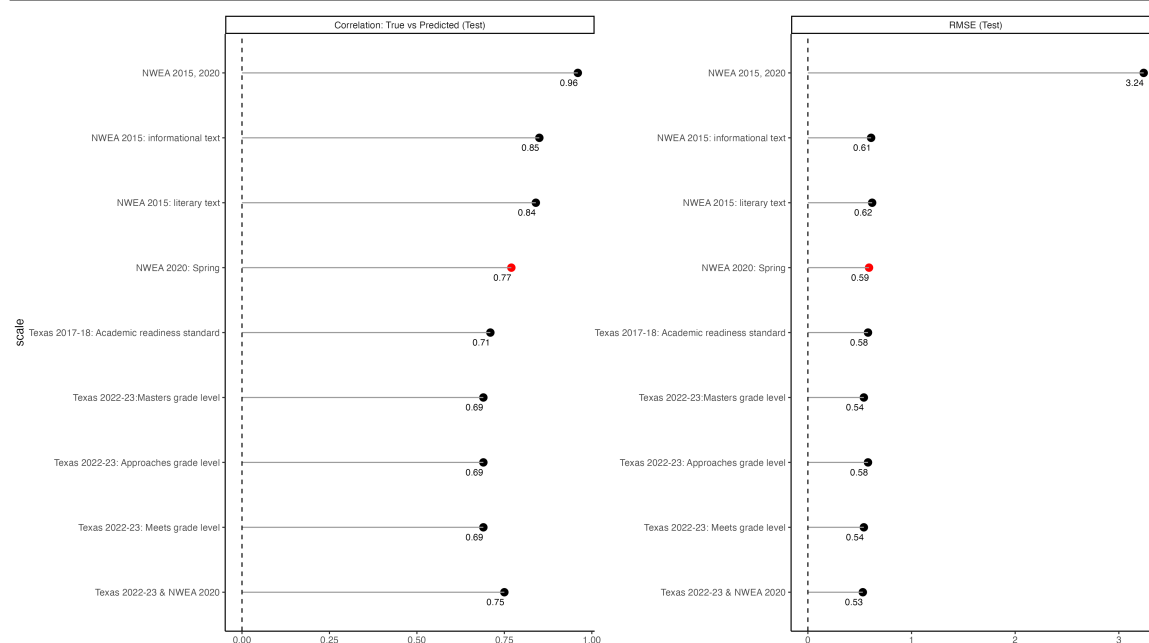


Figure 3: Robustness to Vertical Scale Choice

Note. (Left panel) Correlation of true vs predicted difficulty for test data; (Right panel) RMSE for true vs predicted difficulty. The main results reported in the paper use the NWEA Spring 2020 vertical scale, shown in red in the figure. Note that results reported in the paper are conservative and not extremely sensitive to the choice of vertical scale.

G.2 BERT Embeddings: Robustness to Text Input Format

Table 9: BERT Model: Robustness of Prediction Results

	RMSE		Correlation	
	Train	Test	Train	Test
Main Results: Text input is Question + Correct answer + Wrong answers + Passage				
LLM embeddings: BERT	0.60	0.66	0.76	0.71
Assessment characteristics, text analysis metrics, & BERT embeddings	0.60	0.64	0.76	0.73
Assessment characteristics, text analysis metrics, & PCA on BERT embeddings	0.58	0.64	0.76	0.72
Alternate Results 1: Text input is Question + Correct answer + Wrong answers				
LLM embeddings: BERT	0.65	0.72	0.71	0.63
Assessment characteristics, text analysis metrics, & BERT embeddings	0.60	0.65	0.76	0.73
Assessment characteristics, text analysis metrics, & PCA on BERT embeddings	0.58	0.61	0.76	0.75
Alternate Results 2: Main results with Cosine Similarity Variables				
LLM embeddings: BERT	0.60	0.66	0.76	0.72
Assessment characteristics, text analysis metrics, & BERT embeddings	0.60	0.64	0.76	0.74
Assessment characteristics, text analysis metrics, & PCA on BERT embeddings	0.58	0.63	0.76	0.73

The main results report BERT embeddings generated for a statement that merges question text, correct answer, all wrong answers, and passage (Appendix F). The reading passage was removed from the statement used to generate BERT embeddings for Alternate Results 1 (Table 9). If the passage is relevant to how BERT embeddings capture information about the item, we would expect decline in model performance. We do observe this decline: RMSE increases from 0.66 for Main Results to 0.72 for Alternate Results 1, and correlation decreases from 0.72 to 0.63. However, prediction results are similar when human annotated features are included in the model, and when the model uses PCA on BERT embeddings instead of the full embeddings. This suggests that human annotated features might contain complementary information that compensates model performance. This is also interesting because BERT embeddings are generated using a section of the passage (sentence length is truncated at 512 characters). This suggests that a segment of passage still includes useful information; hence, the main results use embeddings generated with the passage included in the text input to the tokenizer.

We also include new predictor variables: cosine similarity is calculated between the embeddings for correct answer and each of the distractors. Embeddings are generated for the sentence that combines: (1) question text (2) a tag for correct/wrong answer, followed by the correct/wrong answer. Next, we calculate cosine similarity between the embeddings for the correct answer and each of the wrong answers, giving us three cosine similarity variables. These variables are included as predictors in the main model. These results are reported as Alternate Results 2. We note marginal improvement in performance between Main Results and Alternate Results 2, where RMSE decreases from 0.64 to 0.63, and correlation marginally improves from 0.72 to 0.73. This suggests that differences or similarities in BERT embeddings for correct answers and distractors don't explain item difficulty in the context of this dataset.

G.3 Results for Unadjusted p-values

This section presents the results for unadjusted p-values i.e., the p-values reported in the dataset that have not been adjusted for grade-level differences. As discussed in the methods section, this measure of difficulty does not have an inherent meaning. The results show a lower RMSE, likely because grade-level variation has not been accounted for. However, the correlation is also low (in the range of 0.10–0.3).

Table 10: Results for Predicting Item Difficulty: Outcome Variable is Unadjusted p-value

	RMSE		Correlation	
	Train	Test	Train	Test
Results from human annotated features				
State, Grade, Year	0.12	0.12	0.28	0.29
Test features	0.13	0.13	0.15	0.10
Text analysis features	0.12	0.13	0.36	0.11
Results from LLM embeddings only				
Only BERT embeddings	0.11	0.12	0.57	0.30
Only LLaMA embeddings	0.10	0.12	0.69	0.35
Results from LLM embeddings and annotated features				
All features & BERT embeddings	0.10	0.12	0.60	0.32
All features & PCA on BERT embeddings	0.12	0.12	0.46	0.25
All features & LLaMA embeddings	0.10	0.12	0.72	0.35
All features & PCA on LLaMA embeddings	0.12	0.12	0.46	0.25
All features & PCA on ModernBERT embeddings	0.12	0.12	0.45	0.25

Generative-Evaluative Agreement: A Necessary Validity Criterion for LLM-Enabled Adaptive Assessment

Grandee Lee and Yue Wang and Che Yee Lye and Luke Peh

Singapore University of Social Sciences,

463 Clementi Rd, 599494, Singapore

{grandeelee, wangyue, cylie, lukepeh1c}@suss.edu.sg

Abstract

When the same LLM generates assessment items, simulates student responses, and scores them, the validation loop is self-referential. We introduce **Generative-Evaluative Agreement (GEA)**, a validity criterion measuring whether an LLM’s scoring function recovers the skill levels its generative function was instructed to produce. In the first direct measurement of GEA on a two-stage adaptive assessment, the model recovers roughly half the intended variance ($r = 0.698$) with systematic positive bias. GEA is strong ($r > 0.7$) for syntactically verifiable skills but near zero for design-level skills, and low-skill overestimation inflates scores near the routing threshold. We argue that granular, skill-decomposed rubrics are the principal proposed mechanism for strengthening GEA and outline complementary mitigations.

1 Introduction

Computerized adaptive testing (CAT) traditionally relies on item banks pre-calibrated via Item Response Theory (IRT), where every item has known difficulty and discrimination parameters estimated from hundreds of real responses (van der Linden and Glas, 2010). LLM-enabled adaptive assessment disrupts this paradigm: items are generated dynamically, each student potentially receives a unique test, and classical calibration (requiring 50–200 respondents per item; Lord 1980) becomes infeasible. Revisions to rubrics, prompts, or course materials become psychometrically consequential when they change what is being measured, how performance is elicited, how responses are scored, or how scores are interpreted. In these cases, prior calibration and validity may no longer be transportable, and at least part of the item bank may need to be re-authored, relinked, re-calibrated, or revalidated (Han and Guo, 2011). In a typical school setting, learning outcomes are updated when

curricula evolve, rubrics are refined each semester as instructors identify ambiguities, and course restructuring changes the skill prerequisites for each assignment. LLM-based systems absorb these changes through prompt and rubric updates alone, but the validity of the resulting assessment must be re-established each time.

This creates a **bootstrapping problem**: the system cannot be validated without real student data, but cannot be deployed at scale without prior validation. Human review of each generated item is infeasible when the item space is effectively infinite, and the classical validity pipeline (pre-calibrate, validate against human raters, then deploy) does not apply. Simulation-based validation offers a pragmatic alternative. Liu et al. (2024) demonstrated that ensembles of LLM-simulated respondents can approximate human item calibration with correlations exceeding 0.89. Zheng et al. (2026) used Monte Carlo simulation to identify optimal CAT configurations before empirical evaluation. Marquez-Carpintero et al. (2025) reviewed LLM-simulated student profiles for pre-deployment testing of pedagogical systems. However, when the same LLM generates items, simulates student responses, and scores them, the validation loop is self-referential. If the model’s representation of skill levels is inconsistent across its generative and evaluative functions, the system validates itself against a distorted mirror.

This paper introduces **Generative-Evaluative Agreement (GEA)** as the formal criterion for this internal consistency: when an LLM generates a response at an intended skill level, does scoring recover that level? Valid routing decisions require scores that faithfully reflect the intended construct, but “intended difficulty” exists only in the model’s internal representation, accessed through two different computational paths (generation and evaluation) that traverse different prompt-conditioned regions of the same model. Empirical verification of

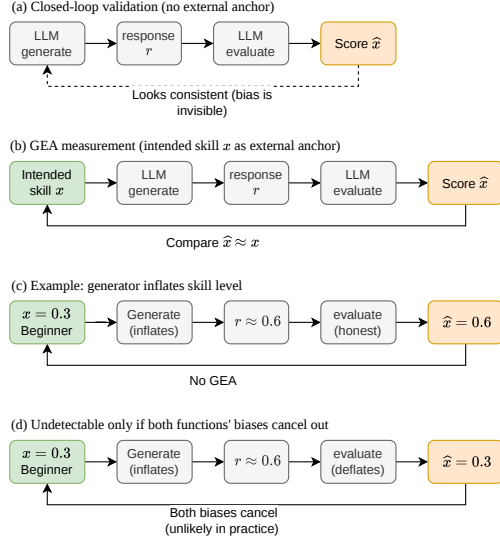


Figure 1: GEA measurement versus closed-loop self-validation. (a) In a pure closed loop, there is no external anchor and bias is invisible. (b) GEA introduces the intended skill level x as an external reference point. (c) When the generator inflates skill, GEA detects the discrepancy. (d) Bias is undetectable only if both functions share the exact same misconception, an unlikely scenario given that generation and evaluation traverse different prompt-conditioned paths.

their alignment is therefore a necessary (though not sufficient) validity condition for any LLM-based adaptive assessment that uses simulation for calibration.

1.1 Definition

Generative-Evaluative Agreement (GEA) is the degree to which an LLM’s generative representation of skill levels is consistent with its evaluative representation. Formally, if the model generates a response $\mathbf{r}$ conditioned on skill level x , then scoring $\mathbf{r}$ should recover x within acceptable error bounds:

$$\mathbb{E}[\text{score}(\mathbf{r}) \mid \mathbf{r} \sim \text{generate}(x)] \approx x \quad (1)$$

Here $x, \text{score}(\mathbf{r}) \in [0, 1]$ are continuous per-skill scores; ordinal proficiency bands (Appendix D) are derived post hoc for reporting.

We operationalise “ $\approx$ ” through two primary metrics: Pearson r for rank-order fidelity and signed bias for systematic directionality. We propose two actionable benchmarks: $r > 0.7$ (strong GEA) to support fine-grained proficiency reporting, and $r > 0.4$ (moderate GEA) to support binary routing decisions. Skills below $r = 0.4$ should not be used for adaptive routing without human validation.

Crucially, GEA measurement is *not* equivalent to closed-loop self-validation. Figure 1 illustrates the distinction. In a pure closed-loop system (panel a), the model generates and scores with no external reference, so any systematic bias is invisible. In GEA measurement (panel b), the intended skill level x serves as an external anchor. If the generator inflates skill (panel c), the discrepancy $\hat{x} \neq x$ reveals the generation bias. GEA can only fail to detect bias when both functions share the *exact same* misconception of x (panel d), which is unlikely given the empirical evidence of divergence reviewed in Section 2.2. Figure 2 shows the concrete assessment architecture in which GEA is measured.

2 Background

2.1 The Closed-Loop Problem

In traditional CAT, item parameters and scoring functions are independently validated against real human response data. In LLM-based adaptive systems, the model performs both roles with no external anchor. An instructive (though imperfect) analogy comes from the generative/discriminative distinction: Ng and Jordan (2001) showed that models learning $P(X \mid Y)$ and $P(Y \mid X)$ can disagree at finite capacity. In an LLM, both generation and evaluation share the same weights, but are conditioned on different prompts that traverse different computational paths. Generation is dominated by fluency priors; evaluation by criterion matching. Shared architecture makes alignment *plausible* but does not *guarantee* it (Oh et al., 2024; West et al., 2023).

2.2 Empirical Evidence of Divergence

LLMs struggle to simulate lower-proficiency cognitive states (Yuan et al., 2026): expert knowledge leaks through despite skill-level prompting. Srivatsa et al. (2025) tested 11 LLMs against real NAEP data and found no model-prompt pair faithfully reproduced real student distributions; Wu et al. (2025) confirmed this for Python programming. LLMs also systematically rate their own outputs higher than equivalent text from other sources (**self-preference bias**; Panickssery et al. 2024), with the mechanism identified as perplexity-based familiarity (Wataoka et al., 2024). Even proprietary models show low intra-rater consistency at temperature > 0 (Lee et al., 2024b). In simulation-based calibration, these mechanisms compound: the result may appear internally consistent but is not exter-

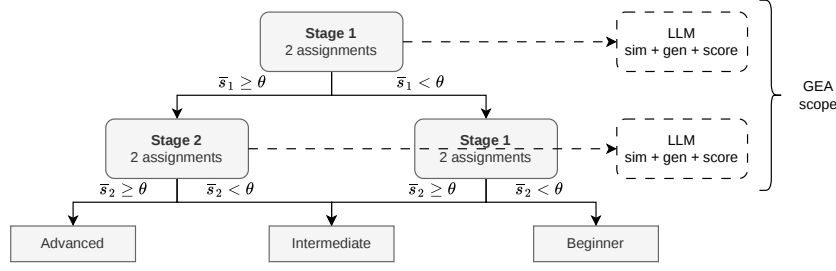


Figure 2: Adaptive assessment flow. The LLM generates assignments and scores responses at each stage. Routing depends on cumulative score $\bar{s}$ crossing threshold θ . GEA measures consistency between the LLM’s generative and evaluative functions.

nally valid.

2.3 Implications for Calibration

When simulation is the only feasible calibration method, GEA becomes the gatekeeper for trustworthiness. If GEA is low, score distributions reflect the model’s self-consistency rather than real student performance. Wang et al. (2025a) argue that Generalizability Theory and Many-Facet Rasch Measurement are needed to decompose multiple simultaneous error sources rather than collapsing them into a single coefficient. Even in real deployment, the generative side affects question generation: if items are at the wrong difficulty, routing decisions are based on mis-targeted items regardless of grading accuracy.

2.4 Related Work

GEA connects to several research threads. The automated essay scoring (AES) literature has studied inter-rater reliability for decades (Ramesh and Dash, 2022); GEA differs in that rater and author are the same model. From a psychometric perspective, GEA instantiates the *substantive* component of Messick’s (1989) construct validity framework, and aligns with the *Standards for Educational and Psychological Testing* (American Educational Research Association et al., 2014) requirement for evidence that scores support intended interpretations. The LLM-as-judge paradigm (Zheng et al., 2024) has documented self-preference and position bias; GEA extends this from evaluation-only settings to the generate-then-evaluate pipeline where generation bias compounds with evaluation bias.

3 Empirical Measurement of GEA

This section presents the empirical measurement of GEA in the sense defined in Section 1.1 for the concrete case of Python object-oriented program-

ming (OOP) coding tasks. The same Claude model performs both code generation and rubric-based evaluation against the 24-skill taxonomy (class definition, inheritance, exception handling, etc.; full list in Appendix B).

3.1 Simulation Design

Student profiles. We generated 150 synthetic student profiles, each comprising a 24-dimensional skill vector $\mathbf{x} \in [0, 1]^{24}$ corresponding to the official learning outcomes. (Table 2). Skills are grouped into four progressive groups: Group A (S01–S08, class basics), Group B (S01–S04, S06–S07, S09–S13, class variables and composition), Group C (S01–S04, S06, S09, S14–S21, inheritance and polymorphism), and Group D (S01, S14–S15, S22–S24, exception handling). For more details, see Appendix B. Profiles were sampled from 10 archetypes (e.g., “Absolute Beginner,” “Lab 2 Proficient,” “Advanced”) with Gaussian noise ($\sigma = 0.04$) to produce realistic within-archetype variation.

Assessment slots. Every student attempted all 6 assignment slots regardless of skill level, bypassing the adaptive routing to ensure full coverage: Each slot tests a designated subset of the 24 skills; non-applicable skills are marked -1.0 in the output vector and excluded from scoring. Scenario entities were assigned deterministically per student (seeded on student ID) to ensure reproducibility.

Generate-then-score protocol. For each (student, slot) pair, we executed two sequential API calls to Claude Sonnet 4.6:

1. **Generate:** Given the student’s full skill profile (24 skill scores with per-skill natural-language descriptors) and the assignment question, the model was prompted to produce

Table 1: Overall GEA statistics (7,788 paired skill observations, 150 students, 23 skills). 95% bootstrap CIs from 1,000 resamples.

Metric	Value	95% CI
Pearson r (pooled)	0.698	[.684, .712]
Mean bias (obs – true)	+0.059	[+.053, +.066]
Exact proficiency match	34.8%	

Python code that *precisely matches* the specified skill levels, including deliberate errors, omissions, and partial implementations for low-scoring skills.

2. **Score:** The generated code was submitted to the same model’s scoring function with the identical rubric, which returned a 24-element observed skill vector $\hat{\mathbf{x}} \in \{-1.0\} \cup [0, 1]^{24}$ and a scalar score $s = \text{round}(\text{mean}(\hat{x}_i : \hat{x}_i \neq -1) \times 100)$.

This yields paired observations $(x_i, \hat{x}_i)$ for every skill i that is applicable in a given slot, providing the raw material for measuring Equation 1.

Scale. All 150 students had completed all 6 slots producing 862 result records and 7,788 paired (true, observed) skill-level observations across 23 of 24 skills (S13, Dictionary Collection Management, was not tested in any slot). Table 1 summarises the aggregate agreement statistics.

4 GEA Findings

The pooled Pearson correlation of $r = 0.698$ indicates that the LLM’s evaluative function recovers roughly half the variance ($R^2 \approx 0.49$) in the true skill levels it was asked to generate. The positive mean bias of +0.059 confirms the direction predicted by self-preference bias: the model systematically overestimates the skill level of its own generated code.

At the proficiency-level granularity used for reporting (8 ordinal levels from *Not Demonstrated* to *Mastered*; boundaries in Appendix D), exact classification accuracy is only 34.8%, rising to 64.4% within ± 1 adjacent level. This represents moderate agreement, sufficient to distinguish broad skill bands but insufficient for fine-grained proficiency reporting.

4.1 Per-Skill GEA

GEA varies dramatically across skills. Table 2 presents the full per-skill breakdown, sorted by Pearson r . After Benjamini-Hochberg correction

Table 2: Per-skill GEA metrics, sorted by Pearson r . Skills marked with $\star$ are mandatory in the assessment rubric.

Skill		n	r	Bias
S05	Setter w/ Validation	290	.88	+.10
S09 $\star$	Class Variable	146	.83	+.26
S23	Raise Exception	145	.80	+.12
S11	Composition	139	.80	+.13
S12	Delegation	139	.80	+.17
S10	Class Var Mod	146	.78	+.14
S18 $\star$	Override Refine	285	.77	+.13
S24	Try/Except	145	.77	+.11
S07	Compute Method	429	.72	+.04
S15	super().__init__	433	.71	+.08
S06	__str__	717	.65	−.04
S01	Class Def	848	.62	+.06
S04	Getter Property	716	.59	+.02
S16	Subclass Attrs	288	.57	+.21
S08	Mutate Method	287	.55	−.06
S22	Custom Exception	145	.55	+.21
S02	Constructor	717	.51	−.04
S14	Subclass Def	433	.47	+.25
S03 $\star$	Private Vars	717	.43	± 0.0
S19	ABC Definition	114	.08	−.10
S17 $\star$	Override Replace	288	.06	+.19
S21	Polymorphism	113	−.03	−.07
S20	Concrete Subclass	108	n/a	−.09

for multiple comparisons across 23 skills, 19 correlations remain significant at $\alpha = 0.05$; the four non-significant skills are precisely the near-zero GEA tier. Three tiers emerge:

Strong GEA ($r > 0.7$; 10 skills). Skills with concrete, syntactically verifiable indicators, such as setter validation logic (S05, $r = 0.88$), class variable declaration with underscore convention (S09, $r = 0.83$), exception raising (S23, $r = 0.80$), and composition via object attributes (S11, $r = 0.80$), show the strongest agreement. These skills have unambiguous code signatures: a `@attr.setter` with a conditional check, a `_count` class variable, a `raise CustomError()` statement. The rubric criteria map directly to syntactic patterns that both the generator and evaluator can reliably target.

Moderate GEA ($0.4 < r < 0.7$; 9 skills). Foundational skills tested across many slots (S01, S02, S04, S06) show moderate agreement ($r = 0.43$ – 0.65). These skills are near-universal in student code (almost every submission defines a class, writes a constructor, uses properties), creating a ceiling effect that compresses variance. Subclass-related skills (S14, S16) show moderate r but high bias (+0.21 to +0.25), indicating the generator systematically over-performs on inheritance tasks relative to the intended skill level.

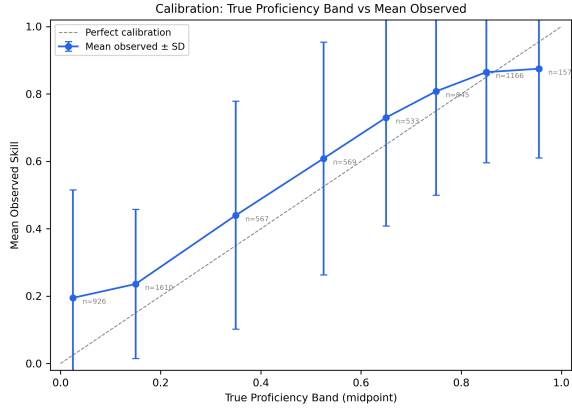


Figure 3: Calibration curve: mean observed skill as a function of true proficiency band. Error bars show ± 1 SD. The dashed diagonal represents perfect calibration. The curve lies above the diagonal at low skill levels (overestimation) and converges at high levels.

Near-zero GEA ($r < 0.1$; 4 skills). Abstract design skills, including ABC definition (S19, $r = 0.08$), method overriding by replacement (S17, $r = 0.06$), polymorphism (S21, $r = -0.03$), and concrete subclass implementation (S20, constant output), show essentially no correlation between intended and observed skill levels. S20 is degenerate: the evaluator assigned nearly identical scores regardless of the true skill level, collapsing all variation. These skills require *design decisions* (choosing to define an ABC, choosing to use polymorphic dispatch) rather than *syntactic patterns*, making them harder for the generator to “partially implement” and harder for the evaluator to grade on a continuous scale.

This pattern (strong GEA for syntactically verifiable skills, weak GEA for design-level skills) is consistent with the scope limitation noted in Section 6: code assessment benefits from partially verifiable ground truth, but that benefit is concentrated in the syntactic stratum of the skill taxonomy. Design-level skills behave more like the subjective assessment domains where GEA is expected to be weakest.

4.2 Calibration and Bias Structure

The calibration curve (Figure 3) reveals an asymmetric bias:

- **Low-skill overestimation:** Students in the “Not Demonstrated” band (true ≈ 0.0) receive mean observed scores of ≈ 0.20 , a $+0.20$ bias. The LLM struggles to generate authentically incompetent code; even when instructed to

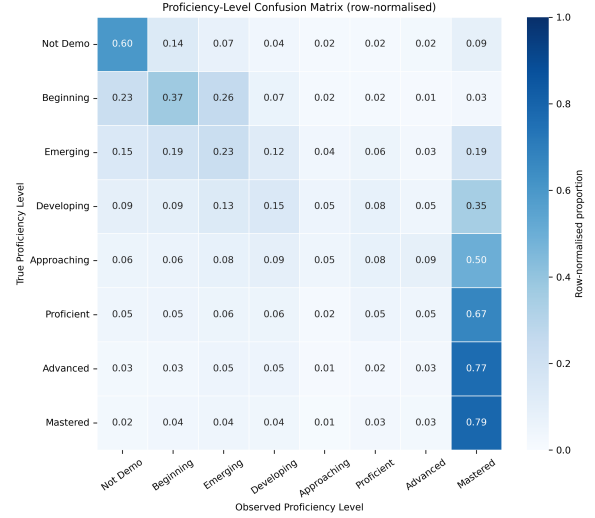


Figure 4: Row-normalised confusion matrix mapping true proficiency levels (rows) to observed proficiency levels (columns). The strong rightward bias in the “Approaching” through “Mastered” rows indicates systematic upward misclassification at higher skill levels.

omit a skill, the generated code tends to include partial or vestigial implementations that the evaluator detects.

- **High-skill convergence:** Students in the “Advanced” and “Mastered” bands (true > 0.80) are scored close to their true levels, with the curve approaching the diagonal. The model finds it easier to generate competent code and to recognise competence.
- **Mid-range compression:** The “Developing” through “Proficient” bands (0.45–0.80) show the highest variance (largest error bars), suggesting the model has difficulty maintaining fine-grained distinctions in the middle of the skill range.

This asymmetry has direct implications for adaptive routing. The system’s routing threshold (≥ 50 for High path) operates in the mid-range where calibration is poorest. The upward bias at low skill levels means that weak students are systematically overscored, making them more likely to exceed the threshold and be routed to the High path, exactly the misrouting scenario described in Section 6.

4.3 Proficiency-Level Classification

The confusion matrix (Figure 4) reveals the proficiency-level consequences of the biases identified above.

Upward collapse to “Mastered.” The dominant pattern is a strong rightward shift: students at “Approaching” (true 0.60–0.70) are classified as “Mastered” 50% of the time; “Proficient” (true 0.70–0.80) collapses to “Mastered” 67% of the time; “Advanced” (true 0.80–0.90) reaches 77%. The evaluator effectively treats any competent implementation as “Mastered,” failing to distinguish gradations within the upper half of the skill range. This is a direct manifestation of **self-preference bias**: the model’s own generated code, even when deliberately degraded, retains low perplexity from the evaluator’s perspective, inflating scores toward the ceiling.

Low-skill recovery. The “Not Demonstrated” level is the best-recovered category at 60% exact match, consistent with the observation that the absence of a skill (e.g., no `@property` defined at all) is a binary, syntactically verifiable condition. However, even here, 14% of truly absent skills are scored as “Beginning” and 9% as “Mastered,” the latter likely reflecting cases where the generator failed to suppress the skill despite being instructed to (competence paradox leakage).

Implications for proficiency reporting. The 8-level proficiency scale is not reliably recoverable through the generate-then-score pipeline. A coarser 3- or 4-level scale would better match the system’s effective resolution. For adaptive routing (threshold at 50), the upward bias at low skill levels ($+0.059 \times 100 \approx 6$ points) inflates Stage 1 scores for weak students, increasing misrouting probability to the High path.

5 Strengthening GEA

The empirical findings above reveal that GEA is partial and skill-dependent. This section examines the mechanisms available to strengthen it, beginning with the most impactful (granular rubrics) and progressing to complementary strategies that address failure modes rubrics alone cannot resolve.

5.1 Rubrics as Constrained Reasoning Paths

Section 2.1 identified the core architectural problem: generation and evaluation traverse different computational paths through the model with no guarantee of alignment. Granular rubrics address this by providing a **shared external specification** that constrains both paths to pass through the same intermediate representation, namely the rubric’s criteria.

Wang et al. (2025b) formalise this as a graph reasoning problem. Without rubrics, the path from task description to score can traverse arbitrary reasoning states. With rubrics, both generation (“produce code demonstrating these specific skills”) and evaluation (“check for these specific skills”) are forced through the same intermediate checkpoints. Under the assumption that aligning these intermediate nodes increases the overlap between the model’s generative and evaluative reasoning paths, rubric-guided assessment should exhibit higher GEA than holistic assessment.

In the context of this system, the rubric defines per-assignment skill vector tables specifying exactly which of the 24 skills are applicable to each assignment and what constitutes mastery. This provides structural decomposition: both the question generator and the response evaluator reference the same granular criteria, reducing the degrees of freedom available for the two paths to diverge. The three-tier GEA pattern in Table 2 provides indirect evidence for this mechanism: skills with unambiguous syntactic rubric criteria (e.g., “code contains a `@property` decorator”) show strong GEA, while skills with holistic criteria (e.g., “demonstrates appropriate use of polymorphism”) show near-zero GEA.

5.2 Empirical Support from the Literature

Rubric quality directly predicts scoring accuracy. Wu et al. (2024) found a Spearman rank correlation of $\rho = 0.94$ ($p < 0.01$) between analytic rubric alignment (with human-crafted rubrics) and automated scoring accuracy. Without rubrics, accuracy was 34.8%; with human-crafted analytic rubrics, 50.4%; with LLM-generated rubrics guided by holistic rubrics, 54.6%. The near-perfect correlation between rubric quality and scoring performance establishes that the evaluative path is strongly anchored by rubric specification. For GEA, this implies that the evaluative side of the agreement can be substantially improved through rubric design alone, without changing the model.

Rubric-aligned component extraction yields consistent gains. AutoSCORE’s two-agent design (first extract rubric-relevant components into a structured JSON representation, then score based on those components) improved QWK by up to 37% (Essay Set, GPT-4o) and 74% (Science, LLaMA-8B) over single-agent baselines (Wang et al., 2025b). Gains were largest on complex, multi-dimensional rubrics, which parallels this sys-

tem’s 24-skill vector with per-assignment coverage tables. The framework demonstrates that decomposing evaluation into criterion-level checks before holistic scoring reduces rubric misalignment and evaluator shortcuts. This system’s architecture already follows this pattern: the evaluator produces a 24-element skill vector before computing a scalar score.

Rubrics balance accuracy across proficiency levels. Lee et al. (2024a) found that Chain-of-Thought prompting combined with scoring rubrics yielded a 13.4% accuracy increase and, critically, more balanced accuracy across different proficiency categories. Without rubrics, LLMs were biased toward certain score ranges; with rubrics, scores distributed more evenly across levels. This directly addresses the calibration asymmetry observed in Figure 3: rubric-guided evaluation may reduce the systematic overestimation at low skill levels that currently inflates routing scores.

Rubrics prevent evaluator shortcuts. Wu et al. (2024) discovered that providing graded student responses (without rubrics) actually *degraded* rubric alignment; the LLM found superficial keyword shortcuts instead of following the intended reasoning chain. Analytic rubrics prevented this by requiring criterion-level assessment, forcing the model through the intended reasoning path rather than surface-level pattern matching. This finding is particularly relevant to GEA: without rubrics, the evaluator may assign high scores based on surface features (e.g., code length, presence of class definitions) rather than the specific skill indicators the generator was instructed to produce or omit.

Rubrics are therefore a **necessary but not sufficient** condition for GEA. They are the most practical strategy within a single-model system, but complete GEA assurance requires complementary measures to address failure modes that rubric design alone cannot resolve.

5.3 Design Principles for GEA-Strengthening Rubrics

Based on the findings above, rubrics that maximise GEA should:

1. **Decompose the construct into discrete, independently assessable skills**, each with a binary or ordinal mastery indicator (as in the 24-skill vector).
2. **Specify per-assignment applicability**: explicitly mark which skills are assessed and

which are not applicable, so neither generation nor evaluation drifts into unintended territory.

3. **Define decision boundaries, not just level descriptions**: specify what distinguishes mastery from non-mastery for each skill, not just what each level “looks like” holistically.
4. **Use structured output formats**: require the evaluator to produce criterion-level judgements (e.g., JSON skill vectors) before aggregating to a holistic score, following AutoSCORE’s component-extraction-then-scoring paradigm (Wang et al., 2025b).
5. **Avoid excessive verbosity**: structural clarity outperforms exhaustive description; long rubrics can degrade performance in some models (Yoshida, 2025).

5.4 Complementary Mitigations

While granular rubrics are the primary strategy for strengthening GEA, they cannot address all failure modes. The competence paradox (the generator producing overly competent code despite low-skill instructions) and self-preference bias (the evaluator inflating scores for model-generated text) require complementary interventions.

Cross-model evaluation. The most direct intervention is using a different model family to score responses than the one that generated them (e.g., generate with Claude, score with GPT-4). This breaks the self-preference loop whereby a model inflates scores for its own low-perplexity outputs (Panickssery et al., 2024). The perplexity-based mechanism identified by Wataoka et al. (2024) implies that cross-model evaluation should be most beneficial for the upper proficiency levels where self-preference bias is strongest (Figure 4). Where cross-model evaluation is impractical due to cost or API constraints, **multi-sample scoring** offers a partial substitute: scoring each response multiple times at non-zero temperature and flagging high-variance items for human review surfaces the stochastic inconsistency that single-pass scoring conceals (Korthals et al., 2026).

Epistemic state specification. On the generation side, the competence paradox can be mitigated by moving beyond naive role-prompting (“act as a beginner”). Yuan et al. (2026) propose using structured misconception inventories and knowledge component graphs that constrain the generative

path to produce behaviourally realistic responses. Rather than asking the model to simulate a general proficiency level, the prompt specifies which knowledge components the student has and has not acquired, which common misconceptions are active, and which error patterns should appear. This converts a vague instruction (“produce beginner code”) into a concrete specification that the generator can follow more reliably, directly improving the generative side of GEA.

Real-student pilot validation. Simulation-derived thresholds should be validated against a small real-student cohort before operational use, following standard psychometric practice. Even a pilot of 10–20 students on a subset of skills would provide an external anchor against which to calibrate the simulation’s bias estimates. This is particularly important for the routing threshold, where the +6-point upward bias identified in Section 4.2 may require adjustment before deployment.

6 Discussion

Threshold sensitivity. We swept the Stage 1 routing threshold θ from 30 to 70 (Table 3), re-routing 140 students with complete data (10 excluded due to incomplete Stage 2 records).

θ	Flip%	Adv%	Int%	Beg%	Mis%
30	10.7	66.4	26.4	7.1	45.0
40	7.1	43.6	45.0	11.4	41.4
50	0.0	24.3	60.0	15.7	34.3
60	5.7	15.7	62.1	22.1	28.6
70	18.6	8.6	56.4	35.0	18.6

Table 3: Threshold sensitivity sweep. Flip%: routing changes vs. baseline $\theta=50$. Mis%: fraction misaligned with true archetype ability.

A **stability plateau** spans $\theta \in [45, 55]$ (<5% flips). Misclassification decreases monotonically as θ rises (45% at $\theta=30$ to 19% at $\theta=70$) because the positive scoring bias pushes observed scores above true ability. However, at $\theta=70$ nearly one in five students would be reclassified. The baseline $\theta=50$ represents a pragmatic compromise within the stability plateau.

Decomposing GEA failure. When GEA is low, the deficit could stem from the generator, the evaluator, or both. For syntactically verifiable skills (e.g., S05, $r = 0.88$), unambiguous code signatures constrain both paths. For design-level skills (e.g., S21, $r = -0.03$), the generator likely over-produces

while the evaluator lacks binary markers to assess degree. Isolating each component requires human scoring of generated code or human-written code at specified skill levels.

Domain dependence. Code assessment occupies a privileged position because programming tasks have partially verifiable ground truth: a class either defines a @property or it does not. Subjective domains (essay argumentation, creative writing) lack these anchors, so GEA findings here likely represent an **upper bound**. The rubric-as-config architecture is domain-agnostic, but empirical GEA guarantees are domain-specific.

Model scaling. We repeated the full simulation using Haiku 4.5 with identical rubrics and profiles (Table 4).

Metric	Haiku 4.5	Sonnet 4.6
Signed bias (obs–true)	+0.31	+0.06
Pearson r (record-level)	0.64	0.92
Terminal Advanced (%)	88.0	24.1

Table 4: Model scaling comparison. Pearson r is computed at the *record level* ($n = 862$ assignment records), which averages across ~ 9 skills per record and is therefore higher than the pooled skill-level $r = 0.698$ in Table 1.

Haiku inflates scores by +17.6 points on average, scoring Absolute Beginners at 44.8 (vs. Sonnet’s 19.8), above the routing threshold, and assigning 88% of students to Advanced. The pooled r difference (Sonnet 0.698 vs. Haiku 0.447) is significant (Fisher $z = 23.8$, $p < 10^{-100}$). GEA is therefore **scale-dependent**: self-preference bias and the competence paradox are amplified at smaller scale.

7 Conclusion

We introduced Generative-Evaluative Agreement (GEA) as a necessary validity criterion for LLM-enabled adaptive assessment. Using 150 synthetic profiles on a Python OOP assessment with Claude Sonnet 4.6, the model recovers roughly half the intended skill variance ($r = 0.698$, 95% CI [.684, .712]) with systematic positive bias, and GEA is strongly skill-dependent: high for syntactically verifiable skills, near zero for design-level skills.

Limitations

The reported GEA estimates should be read with several scoping constraints in mind. All 150 profiles are LLM-sampled rather than drawn from real

students; authentic learner errors likely differ from simulated ones, so a pilot of 10–20 real students would provide an external anchor for the bias estimates and a check on the simulated-error distribution. We evaluate two Claude models (Sonnet 4.6, Haiku 4.5) on Python OOP code only; cross-family replication (e.g., GPT-4o, Gemini, open-weight models) and subjective domains that lack the partial verifiability of code (essays, open-ended argumentation) would test whether the strong-vs-weak-GEA stratification we observe generalises beyond this setting. The rubric-decomposition argument and complementary mitigations (cross-model scoring, epistemic-state specification, multi-sample evaluation) are supported by prior work and by the per-skill GEA contrast in Table 2, but we do not directly ablate rubric granularity or scorer identity on the same task; comparing holistic against decomposed rubrics, and same-model against cross-model scoring, is the most immediate empirical follow-up. Finally, GEA can in principle fail to detect bias when generator and evaluator share an identical distortion—for example, a sycophantic scorer that deflates an inflated generator’s score to match user expectations—and precisely quantifying this residual risk requires human-scored anchor responses to decompose GEA failure between the two functions (Section 6). The intended skill level x nonetheless remains an external reference that distinguishes GEA measurement from pure closed-loop self-validation, and GEA provides a concrete, measurable criterion any LLM-based assessment system can report before deployment.

References

- American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. 2014. *Standards for Educational and Psychological Testing*. American Educational Research Association, Washington, DC.
- K. T. Han and F. Guo. 2011. Potential impact of item parameter drift due to practice and curriculum change on item calibration in computerized adaptive testing. GMAC Research Report RR-11-02, Graduate Management Admission Council.
- Luke Korthals, Emma Akrong, Gali Geller, Hannes Rosenbusch, Raoul Grasman, and Ingmar Visser. 2026. [Towards reliable LLM grading through self-consistency and selective human review: Higher accuracy, less work](#). *Machine Learning and Knowledge Extraction*, 8(3):74.
- Gyeong-Geon Lee, Ehsan Latif, Xuansheng Wu, Ninghao Liu, and Xiaoming Zhai. 2024a. [Applying large language models and chain-of-thought for automatic scoring](#). *Preprint*, arXiv:2312.03748.
- Noah Lee, Jiwoo Hong, and James Thorne. 2024b. [Evaluating the consistency of LLM evaluators](#). *Preprint*, arXiv:2412.00543.
- Yunting Liu, Shreya Bhandari, and Zachary A. Pardos. 2024. [Leveraging LLM-respondents for item evaluation: a psychometric analysis](#). *Preprint*, arXiv:2407.10899.
- Frederic M. Lord. 1980. *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Luis Marquez-Carpintero, Alberto Lopez-Sellers, and Miguel Cazorla. 2025. [Simulating students with large language models: A review of architecture, mechanisms, and role modelling in education with generative AI](#). *Preprint*, arXiv:2511.06078.
- Samuel Messick. 1989. Validity. In Robert L. Linn, editor, *Educational Measurement*, 3rd edition, pages 13–103. American Council on Education and Macmillan, New York.
- Andrew Y. Ng and Michael I. Jordan. 2001. [On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes](#). In *Advances in Neural Information Processing Systems*, volume 14. MIT Press.
- Juhyun Oh, Eunsu Kim, Inha Cha, and Alice Oh. 2024. [The generative AI paradox on evaluation: What it can solve, it may not evaluate](#). *Preprint*, arXiv:2402.06204.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. [LLM evaluators recognize and favor their own generations](#). *Preprint*, arXiv:2404.13076.
- Dhananjay Ramesh and Sandeep Kumar Dash. 2022. [An analysis on the state of automated essay scoring](#). *Preprint*, arXiv:2205.04083.
- KV Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2025. [Can LLMs reliably simulate real students’ abilities in mathematics and reading comprehension?](#) *Preprint*, arXiv:2507.08232.
- Wim J. van der Linden and Cees A. W. Glas. 2010. *Elements of Adaptive Testing*. Statistics for Social and Behavioral Sciences. Springer, New York.
- Yuehan Wang, Jinyan Huang, Lun Du, Yuxin Guo, Ying Liu, and Rong Wang. 2025a. [Evaluating large language models as raters in large-scale writing assessments: A psychometric framework for reliability and validity](#). *Computers and Education: Artificial Intelligence*, 9:100481.
- Yun Wang, Zhaojun Ding, Xuansheng Wu, Siyue Sun, Ninghao Liu, and Xiaoming Zhai. 2025b. [AutoSCORE: Enhancing automated scoring with multi-agent large language models via structured component recognition](#). *Preprint*, arXiv:2509.21910.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. [Self-preference bias in LLM-as-a-judge](#). *Preprint*, arXiv:2410.21819.

Peter West, Ximing Lu, Nouha Dziri, Faeze Brahman, Linjie Li, Jena D. Hwang, Liwei Jiang, Jillian Fisher, Abhilasha Ravichander, Khyathi Chandu, Benjamin Newman, Pang Wei Koh, Allyson Ettinger, and Yejin Choi. 2023. [The generative AI paradox: “what it can create, it may not understand”](#). *Preprint*, arXiv:2311.00059.

Tao Wu, Jingyuan Chen, Wang Lin, Mengze Li, Yumeng Zhu, Ang Li, Kun Kuang, and Fei Wu. 2025. [Embracing imperfection: Simulating students with diverse cognitive levels using LLM-based agents](#). *Preprint*, arXiv:2505.19997.

Xuansheng Wu, Padmaja Pravin Saraf, Gyeong-Geon Lee, Ehsan Latif, Ninghao Liu, and Xiaoming Zhai. 2024. [Unveiling scoring processes: Dissecting the differences between LLMs and human graders in automatic scoring](#). *Preprint*, arXiv:2407.18328.

Lui Yoshida. 2025. [Do we need a detailed rubric for automated essay scoring using large language models?](#) *Preprint*, arXiv:2505.01035.

Zhihao Yuan, Yunze Xiao, Ming Li, Weihao Xuan, Richard Tong, Mona Diab, and Tom Mitchell. 2026. [Towards valid student simulation with large language models](#). *Preprint*, arXiv:2601.05473.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. [Judging LLM-as-a-judge with MT-Bench and Chatbot Arena](#). *Preprint*, arXiv:2306.05685.

Tianpeng Zheng, Zhehan Jiang, Jiayi Liu, and Shicong Feng. 2026. [Leveraging computerized adaptive testing for cost-effective evaluation of large language models in medical benchmarking](#). *Preprint*, arXiv:2603.23506.

A Assessment Architecture

The system is a conversational assessment tool delivered via Telegram Bot that conducts adaptive, scenario-based coding assessments. Students work through multiple progressive mini coding assignments per stage, evaluated by Claude AI against predefined rubrics. The system routes students to a terminal difficulty level based on cumulative stage performance.

Two-stage adaptive routing. All students begin with Stage 1 (2 assignments covering class basics and composition). Based on their cumulative Stage 1 score relative to threshold θ , students

are routed to either the *High Performer* path (inheritance and exception handling) or the *Low Performer* path (reinforcement of foundational skills). After Stage 2, a terminal level is assigned:

- **Advanced:** Stage 2 High path, cumulative $\geq \theta$
- **Intermediate:** Stage 2 High path, cumulative $< \theta$; or Stage 2 Low path, cumulative $\geq \theta$
- **Beginner:** Stage 2 Low path, cumulative $< \theta$

Dynamic question generation. Questions are not pre-stored. Claude generates each assignment dynamically at runtime based on the rubric, the current stage/path, and a scenario template. Each student receives a slightly different variation of the same assignment—same core objective and difficulty, but different scenario entities (e.g., bank account, airline, cinema), discouraging copying. Every assignment begins with an ASCII UML class diagram showing the class(es) the student must implement.

Scoring. Each submission is evaluated by Claude against the rubric for that assignment. The model returns a 24-element skill vector $\hat{\mathbf{x}} \in \{-1.0\} \cup [0, 1]^{24}$ (where -1.0 denotes inapplicable skills) and a scalar score $s = \text{round}(\text{mean}(\hat{x}_i : \hat{x}_i \neq -1) \times 100)$. The scalar score drives routing; the skill vector provides diagnostic detail.

B Skill Taxonomy

The 24 skills are organised into four progressive groups corresponding to the course’s lab sequence. Skills marked with $\star$ are mandatory anchors in the assessment rubric.

C Assessment Slot–Skill Coverage

Table 6 shows which skills are assessed (scored 0.0–1.0) in each of the 6 assessment slots. Skills not listed for a slot receive -1.0 (not applicable) in the skill vector.

D Proficiency Scale

Continuous skill scores are mapped to ordinal proficiency levels using the boundaries in Table 7. These levels are used for the confusion matrix analysis (Section 4.3) and for natural-language descriptors in student profiles.

E Student Profile Archetypes

150 synthetic student profiles were sampled from 10 archetypes (Table 8). Each archetype defines per-group skill ranges from which individual skill scores are drawn uniformly, with Gaussian noise ($\sigma = 0.04$) added to produce within-archetype variation. Profiles were generated with a fixed random seed for reproducibility.

Example profile. Table 9 shows a representative “Lab 2 Developing” profile. The student has strong class basics (Group A, mean 0.84), partial composition skills (Group B, mean 0.46), and minimal inheritance/exception knowledge (Groups C–D, < 0.10). Each skill carries a natural-language descriptor (shown for selected skills) that is included in the generation prompt to constrain the LLM’s code output.

F Prompt Templates

F.1 Code Generation Prompt (Simulation)

For each (student, slot) pair, the following prompt is sent to Claude Sonnet 4.6 to generate code that matches the student’s skill profile. The STUDENT SKILL PROFILE block lists only the skills applicable to the current slot, with each skill’s numeric score, proficiency level, and natural-language descriptor.

You are simulating a student submitting a Python OOP coding assignment.

The student’s exact skill profile for the skills tested in this assignment is provided below. Write Python code that a student at PRECISELY these skill levels would produce. Faithfully reflect each described weakness and strength – do not average them out or homogenise the code.

STUDENT SKILL PROFILE (relevant skills only):

- S01 Class Definition: 0.82 (Advanced) – Class is well-defined, correctly named, and structurally complete with multiple methods.
- S03 Private Instance Variables: 0.70 (Proficient) – self.__attr used for key attributes with property getters; not all attributes are private.
- ...

ASSIGNMENT:

{the generated assignment text}

Rules:

1. Output Python code only – no explanations, no markdown fences, no preamble.
2. For skills rated “Not Demonstrated” or “Beginning”, the code must clearly exhibit the described gap.

3. For skills rated “Advanced” or “Mastered”, that aspect of the code must be correct and complete.

4. Each skill reflects its own level independently – the code can be strong in one area and weak in another.

5. Use realistic student-style naming and formatting consistent with the described skill levels.

F.2 Scoring Prompt (Evaluation)

The scoring function sends the full rubric document (including per-assignment skill vector tables with scoring guidance) along with the student’s submission. The prompt instructs the model to:

You are a coding assessment scorer for a Python OOP course.

The full rubric is below.

RUBRIC:
{full RUBRICS.md content}

–

Score the student’s submission for:

- Stage: {stage} - Path: {path} - Assignment: {n} of 2

Assignment given to the student:

""{question text}""

Student’s submission:

""{student code}""

Instructions:

1. Locate the rubric section for this stage, path, and assignment number.
2. Fill in the 24-element skill_vector exactly as defined in the Skill Vector table.
- Use -1.0 for skills marked -1.0 (not applicable).
- Use a float 0.0–1.0 for all other skills, following the scoring guidance.
- Use intermediate values (e.g. 0.3, 0.7) freely.
3. Compute score = round(mean(v_i for v_i if v_i != -1.0) * 100).
4. Write 2–4 sentences of constructive feedback.

Return ONLY a valid JSON object:

{“score”: <int>, “feedback”: “<text>”, “skill_vector”: [<s01>, ..., <s24>]}

F.3 Question Generation Prompt

The question generator receives the full rubric and is instructed to locate the correct section for the given stage, path, and assignment number, substitute the student’s scenario entity into the template, produce the class diagram followed by coding instructions, and return only the assignment text the student will see. Scenario entities are drawn from pre-defined lists:

- **Stage 1:** bank account, airline booking, cinema, grade tracker, event planner, inventory, point-of-sale, membership

- **Stage 2 High:** banking hierarchy, airline, vehicle fleet, cinema chain, course catalogue, training programme
- **Stage 2 Low:** contact book, recipe manager, budget tracker, shopping list, event log, book collection

Entities are assigned deterministically per student (seeded on student ID) to ensure reproducibility across simulation restarts.

G Rubric Excerpts: Strong vs. Weak GEA Skills

To illustrate the rubric specificity hypothesis (Section 5.1), we present the scoring guidance for a strong-GEA skill and a near-zero-GEA skill as they appear in the rubric’s skill vector tables.

S05 — Setter with Validation ($r = 0.88$, **strong GEA**). The rubric specifies concrete syntactic markers at each score level:

Scoring guidance: 1.0 = @attr.setter with a meaningful validation rule; 0.5 = setter present but no validation logic; 0.0 = absent.

The 8-level proficiency descriptors used in student profiles further constrain generation:

The binary nature of the criterion (setter with validation present or absent) provides an unambiguous code signature that both the generator and evaluator can reliably target.

S21 — Polymorphism via Shared Interface ($r = -0.03$, **near-zero GEA**). The rubric criterion is holistic rather than syntactic:

Scoring guidance: 1.0 = loop/function calls same method on mixed-type list without isinstance checks; 0.5 = loop present but uses type checks; 0.0 = no polymorphic usage.

The corresponding proficiency descriptors:

Polymorphism is a *design choice* (choosing to iterate over a heterogeneous list) rather than a syntactic pattern, making it harder for the generator to “partially implement” and harder for the evaluator to grade on a continuous scale. The descriptor levels describe *degrees of design completeness* rather than presence/absence of identifiable code tokens.

H Routing Logic

The routing state machine operates as follows. Let $\bar{s}_k$ denote the cumulative (mean) score after completing all assignments in stage k .

1. **Stage 1 completion:** Compute $\bar{s}_1$. If $\bar{s}_1 \geq \theta$, route to Stage 2 High path; otherwise Stage 2 Low path.
2. **Stage 2 completion:** Compute $\bar{s}_2$.
 - High path: if $\bar{s}_2 \geq \theta$ then *Advanced*; else *Intermediate*.
 - Low path: if $\bar{s}_2 \geq \theta$ then *Intermediate*; else *Beginner*.

The threshold $\theta = 50$ is a placeholder pending real-student calibration. The threshold sensitivity analysis (Table 3) confirms a stability plateau at $\theta \in [45, 55]$ where <5% of routing decisions change.

Each stage comprises exactly 2 assignments. Scores within a stage are averaged (not summed), so $\bar{s}_k = \frac{1}{2} \sum_{j=1}^2 s_{k,j}$ where $s_{k,j} \in [0, 100]$. Sessions are stored in memory keyed by Telegram user ID and do not survive bot restarts.

Table 5: Complete 24-skill taxonomy with descriptions and demonstration criteria.

ID	Skill	Description	Demonstrated by
Group A — Class Basics (Lab 1)			
S01	Class Definition	Defines a class using <code>class</code> with PascalCase name; non-empty body	Any valid class definition with ≥ 1 method
S02	Constructor Init	Uses <code>__init__(self, ...)</code> to initialise instance attributes	<code>__init__</code> present, ≥ 2 attrs assigned via <code>self</code>
S03*	Private Vars	Declares <code>self.__attr</code> with name mangling	≥ 1 <code>self.__attr</code> with property/getter access
S04	Getter Property	@property decorator exposes private attribute	Method with @property returning <code>self.__attr</code>
S05	Setter w/ Validation	@attr.setter with ≥ 1 validation rule	Setter checks condition before assigning
S06	<code>__str__</code>	Returns human-readable string using instance data	f-string with ≥ 2 instance attributes
S07	Compute Method	Instance method reading <code>self</code> attrs, returns computed value	return using <code>self.attr</code> in calculation
S08	Mutate Method	Instance method modifying <code>self</code> attributes	Assigns new value to <code>self.__attr</code>
Group B — Class Variables & Composition (Lab 2)			
S09*	Class Variable	Shared class-level <code>_attr = value</code> (not in <code>__init__</code>)	<code>_attr</code> at class level, accessed via <code>ClassName._attr</code>
S10	Class Var Mod	Correctly accesses/modifies class variable at runtime	Change reflected across all instances
S11	Composition	Has-a relationship: object stored as attribute (DI)	<code>__init__</code> accepts object param, assigns to <code>self.__other</code>
S12	Delegation	Outer class calls composed object's methods/properties	<code>self.__inner.method()</code> in outer class
S13	Dict Collection	Dictionary manages collection with add/search/remove	<code>dict[key] = obj</code> , <code>dict.get()</code> , <code>dict.pop()</code>
Group C — Inheritance (Lab 3)			
S14	Subclass Def	<code>class Child(Parent):</code> with correct parent	Subclass instance can call parent methods
S15	<code>super().__init__()</code>	Calls <code>super().__init__(...)</code> first in subclass	First statement in child <code>__init__</code>
S16	Subclass Attrs	New attributes added after <code>super()</code> call	≥ 1 <code>self.__new_attr</code> after <code>super()</code>
S17*	Override Replace	Completely replaces parent method (no <code>super()</code>)	Same-name method with entirely new logic
S18*	Override Refine	Calls <code>super().method()</code> then extends result	<code>super().method()</code> + additional logic
S19	ABC Definition	<code>ABC + @abstractmethod</code> to define contract	<code>from abc import ABC, abstractmethod</code>
S20	Concrete Subclass	Implements all abstract methods with meaningful bodies	Subclass instantiates without <code>TypeError</code>
S21	Polymorphism	Same method called on mixed-type list, no type checks	Loop over heterogeneous list, no <code>isinstance</code>
Group D — Exception Handling (Lab 4)			
S22	Custom Exception	Class inheriting from <code>Exception</code>	<code>class MyException(Exception):</code> <code>pass</code>
S23	Raise Exception	<code>raise</code> custom exception with descriptive message	<code>if condition:</code> <code>raise MyException("msg")</code>
S24	Try/Except	<code>try/except</code> catching custom + built-in exceptions	≥ 2 distinct conditions caught with messages

Table 6: Slot-to-skill mapping. Each slot tests a designated subset of the 24 skills; remaining skills are marked –1.0.

Slot	Content	Skills assessed	<i>n</i> skills
Stage 1, A1	Lab 1: classes & properties	S01–S08	8
Stage 1, A2	Lab 2: composition	S01–S04, S06–S07, S11–S12	8
Stage 2 High, A1	Lab 3: inheritance	S01–S04, S06, S09–S10, S14–S21	15
Stage 2 High, A2	Lab 4: exceptions	S01, S14–S15, S22–S24	6
Stage 2 Low, A1	Lab 1 reinforcement	S01–S08	8
Stage 2 Low, A2	Lab 3 introduction	S01–S04, S06, S14–S18	10

Table 7: Proficiency level boundaries.

Level	Score range	Midpoint
Not Demonstrated	[0.00, 0.05)	0.025
Beginning	[0.05, 0.25)	0.15
Emerging	[0.25, 0.45)	0.35
Developing	[0.45, 0.60)	0.525
Approaching	[0.60, 0.70)	0.65
Proficient	[0.70, 0.80)	0.75
Advanced	[0.80, 0.90)	0.85
Mastered	[0.90, 1.00]	0.95

Table 8: Archetype definitions. Each cell shows the [lo, hi] range for uniform sampling of skill scores within that sub-group. Sub-groups: A = S01–S08 (class basics), B = S09–S13 (composition), C₁ = S14–S16 (basic subclass), C₂ = S17–S18 (mandatory overrides), C₃ = S19–S21 (advanced inheritance), D = S22–S24 (exceptions).

Archetype	%	A	B	C ₁	C ₂	C ₃	D
Absolute Beginner	8	.00–.22	.00–.10	.00–.07	.00–.07	.00–.05	.00–.05
Lab 1 Developing	12	.22–.52	.00–.14	.00–.09	.00–.09	.00–.05	.00–.05
Lab 1 Proficient	15	.52–.78	.00–.22	.00–.14	.00–.12	.00–.07	.00–.05
Lab 1 Mastered	12	.78–1.0	.18–.48	.00–.18	.00–.14	.00–.09	.00–.05
Lab 2 Developing	12	.68–1.0	.32–.62	.00–.18	.00–.14	.00–.09	.00–.07
Lab 2 Proficient	10	.78–1.0	.62–.92	.08–.28	.05–.22	.00–.12	.00–.09
Lab 3 Developing	12	.72–1.0	.62–1.0	.28–.62	.22–.58	.00–.22	.00–.14
Lab 3 Proficient	10	.82–1.0	.72–1.0	.58–.88	.52–.85	.08–.38	.00–.18
Lab 4 Developing	5	.78–1.0	.72–1.0	.68–1.0	.62–.92	.18–.58	.28–.62
Advanced	4	.85–1.0	.80–1.0	.78–1.0	.75–1.0	.52–.92	.68–1.0

Table 9: Excerpt from student profile 0114 (Lab 2 Developing archetype, overall score 0.41).

Skill	Name	Score	Level
S01	Class Definition	0.82	Advanced
S03★	Private Vars	0.70	Proficient
S05	Setter w/ Validation	0.92	Mastered
S09★	Class Variable	0.37	Emerging
S11	Composition	0.66	Approaching
S14	Subclass Def	0.15	Beginning
S17★	Override Replace	0.00	Not Dem.
S22	Custom Exception	0.01	Not Dem.

Table 10: S05 proficiency descriptors (excerpt).

Level	Descriptor
Not Dem.	No setter defined; private attributes cannot be updated after construction.
Emerging	Setter present but validation logic is absent; any value is accepted.
Proficient	Setter enforces a clear validation rule and handles the invalid case appropriately; minor gap.
Mastered	Setter enforces thorough validation with appropriate response; all edge cases covered.

Table 11: S21 proficiency descriptors (excerpt).

Level	Descriptor
Not Dem.	No polymorphic usage; objects are type-checked before any method call.
Emerging	A polymorphic call is made but only one subclass type is present.
Proficient	Multiple subclass types in a collection; same method called without type checks; each responds correctly.
Mastered	Polymorphism fully and cleanly demonstrated; shared interface called on mixed-type collection with no type checks and correct output per type.

Evaluating LLM-Generated Formative Feedback for Undergraduate Mathematics Through the Lens of Feedback Theory

Aron Gohr¹ Marie-Amelie Lawn² Kevin Gao² Stephen Heslip² Inigo Serjeant²

¹Independent Researcher ²Imperial College London
aron.gohr@gmail.com m.lawn@imperial.ac.uk

Abstract

Large language models can generate feedback on free-form student writing, but there is still limited evidence on whether such feedback is mathematically correct, pedagogically useful, and aligned with expert judgement across varied undergraduate proof-writing tasks. We evaluate LLM-generated feedback on 65 undergraduate proof-writing exercises using Hattie and Timperley’s feedback framework and a grade-agreement metric, comparing two models (GPT-4.1, GPT-5) under two workflow configurations graded by two independent LLM evaluators. GPT-5 produces higher-quality feedback across all dimensions. A mark-scheme-augmented workflow improves grade correlation with human experts for both models, and precomputed mark schemes allow instructors to audit the system before deployment. While the present work cannot supply evidence of benefit in downstream learning tasks, as no controlled trials suitable to show such were run, we do show automated detection and explanation of errors on a wide variety of typical undergraduate-level proof tasks that is aligned with expert judgement on the mathematical issues. However, providing meaningful self-regulation support and controlled studies with students remain to be done. These results show that feedback theory provides a useful lens for evaluating automated mathematical feedback.

1 Introduction

Proof writing is a core skill in mathematics, yet students entering university typically have little experience with it (Moore, 2016). Frequent, detailed formative feedback is important in the acquisition of this skill, but producing it at scale is labour-intensive and inconsistent across graders. Unlike most high-school mathematics tasks, undergraduate proof exercises require assessing the validity, relevance, and level-appropriateness of open-ended arguments. A student may give a correct solution that is unexpected to the grader; an

incorrect argument may still deserve substantial credit if it contains the main ideas; a small computational error may invalidate the solution if the subsequent argument depends on it; and a mathematically correct proof may still be poor if it relies on machinery outside the course, takes large unnecessary detours, or is poorly written. Recently, large language models (LLMs) have achieved strong performance on mathematical reasoning benchmarks (OpenAI, 2024) and have been trialled as tutors (Jurenka et al., 2024; Miller and DiCerbo, 2024). It has also been shown that at least at a high computational budget, they can be used to grade complex proofs (Ma et al., 2025). Educational applications, however, must produce not only grades but *pedagogical feedback* while operating under tight budget and latency constraints. To our knowledge, LLM-generated feedback quality for free-form undergraduate proof exercises has not been previously studied.

We evaluate LLM-generated feedback on 65 undergraduate proof-writing exercises along four dimensions derived from the Hattie–Timperley feedback framework (Hattie and Timperley, 2007): correctness, task-level clarity, process-level guidance, and self-regulation support. We compare two models (GPT-4.1, GPT-5) under two workflow configurations (a direct baseline and a mark-scheme-augmented pipeline), graded by two independent LLM evaluators, and use grade agreement with human experts as a complementary screening metric. Code, prompts, and data are available at https://github.com/agohr/llm_proof_feedback.

Contributions.

1. We operationalise the Hattie–Timperley framework by deriving a rubric for automated evaluation of LLM-generated mathematical feedback.
2. We show that evaluation using this frame-

work reveals quality differences invisible to grade agreement alone. For instance, grade agreement shows no advantage for GPT-5 over GPT-4.1, despite GPT-5 scoring substantially higher on all feedback dimensions.

3. We stress test our tutoring system in several ways: we show that LLMs can effectively find errors in exemplary solutions of our exercises, we analyse the behaviour of our workflows for questions designed to trick the models, we show that both graders prefer the GPT-5 output despite self-serving bias (Panickssery et al., 2024).

Related work. Intelligent tutoring systems have provided automated mathematical feedback for decades, from Andes (Gertner and VanLehn, 2000) to proof-assistant-based systems like Waterproof (Wemmenhove et al., 2022) and LeanTutor (Patel et al., 2026), but these cannot process natural free-form writing. LLM-based systems can (Miller and DiCerbo, 2024; Jurenka et al., 2024), but exhibit failure modes such as sycophancy and hallucination. LLM-as-judge approaches are widely used in the LLM capabilities literature (Zheng et al., 2023); we adopt a cross-grader design to control for self-preference bias (Panickssery et al., 2024).

Feedback theory. Hattie and Timperley (2007) proposed a model distinguishing four feedback levels: *task* (FT), *process* (FP), *self-regulation* (FR), and *self* (FS). Task-level feedback identifies what was correct or incorrect; process-level feedback addresses strategies and reasoning; self-regulation feedback prompts metacognitive monitoring and planning; self-level feedback (“you are smart”) is generally ineffective. We operationalise this framework as a grading scheme for LLM-generated feedback.

2 Methods

Dataset. We develop a synthetic dataset consisting of 65 question–solution pairs from a first-year transition-to-proof course covering number systems, vector geometry, and calculus. Solutions were authored by experienced students familiar with the course and include deliberate errors. The solution authors were the student co-authors in their second or third year who had taken the course. Given a workload of 21 to 22 problems per student they had to work quickly, and were encouraged to write down partial arguments where they could

not finish a solution, and, where natural, to include plausible errors that would be useful test cases. The dataset is therefore controlled and error-rich, but not claimed to represent the full distribution of real student submissions, which we expect will in any case not be uniform across deployments. Three expert markers (two of the authors, one mathematics PhD student) independently graded each item on a 0–5 rubric; grades were reconciled by manual review by one of the authors, who is a lecturer of the course.

Feedback Generation. We compare two feedback-generation models and two workflow configurations, yielding four conditions:

- **Models:** GPT-4.1 and GPT-5.
- **Workflows:**
 - *Baseline-concise*: the model receives the question and student answer with a 200-word limit and generates feedback directly.
 - *MS-w-example*: a multi-step pipeline that first specialises a generic mark scheme to the question, then generates feedback informed by the specialised mark scheme. This workflow was selected based on its strong performance in grade-agreement pre-evaluations.

GPT-5 and GPT-4.1 were selected for this experiment because they represent modern reasoning and non-reasoning language models, respectively. Baseline-concise represents a simple workflow that just asks the model to provide *short* feedback.¹

Hattie–Timperley Evaluation. Each feedback item is scored on four dimensions, operationalised as rubrics with detailed level descriptors (0–5):

1. **Correctness (D1):** Mathematical accuracy of the feedback content.
2. **Task-level clarity (D2/FT):** Whether the feedback specifically identifies what was correct or incorrect, with reference to the exercise’s goals.
3. **Process-level guidance (D3/FP):** Whether the feedback engages with the student’s reasoning strategies and suggests alternatives.

¹We found in early experiments that directly asking the models for feedback tended to result in feedback that in our judgment was clearly too long.

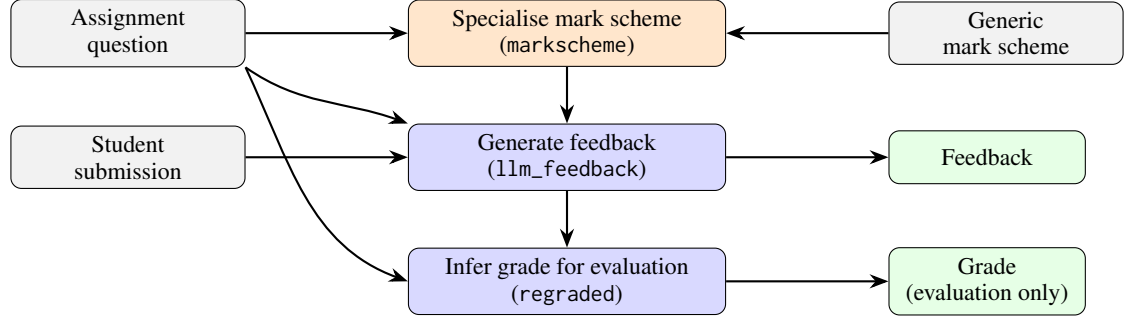


Figure 1: High-level view of the MS-w-example workflow. The mark-scheme step is precomputable because it depends on the assignment and generic rubric, but not on any student submission. In deployment, the system returns only feedback; the grade node is used for evaluation.

Role	Model/settings
Feedback generation	GPT-4.1; GPT-5 with medium reasoning. Defaults otherwise.
HT feedback evaluation	GPT-5.4 and GPT-4.1 as independent graders.
Grade screening	GPT-4.1-nano infers a grade from question and feedback only.
Workflow selection	pre- Earlier grade-correlation sweeps used GPT-4o workflows and GPT-4o grading.

Table 1: Compact model-role summary. Reasoning models do not expose a temperature parameter; non-reasoning calls used temperature zero. We note that OpenAI models do not behave deterministically even at temperature zero.

4. **Self-regulation support (D4/FR):** Whether the feedback prompts self-monitoring, identifies next steps, and attributes outcomes to controllable factors.

We distinguish two roles: *feedback generation* (the model producing feedback for students) and *meta-evaluation* (the model scoring that feedback). Two LLM meta-evaluators, GPT-5.4 and GPT-4.1, independently score all $4 \times 65 = 260$ feedback instances on all four dimensions. For instance, GPT-5, which acts as a feedback-generation model in one experimental condition, is thus evaluated by GPT-5.4 in its meta-evaluator role. Each meta-evaluator first produces a brief chain-of-thought justification and then assigns an integer score, yielding paired scores that support cross-grader reliability analysis. The justifications were retained for human inspection and are part of our code and data repository.

Grade Screening. As a complementary metric, we regrade each feedback instance using a lightweight model (GPT-4.1-nano) that infers a 0–5

grade from the question and feedback alone (student solution hidden), and compute Kendall τ with human grades. The use of a *lightweight* grader is motivated here by two points. First, feedback should be as unambiguous as possible. Second, a mathematically capable grader could correctly set a grade based on feedback that is completely incompetent but leaks enough information about the original submission.

Manually Checking a Sample To check whether GPT-5.4 was grading the model feedback items too generously, we manually graded all 65 question/answer/feedback triples from the MS-w-example workflow with GPT-4.1 as the feedback-generation model. On this sample, LLM ratings were on average *more critical* than human ratings across all four HT categories, except for the GPT-4.1 meta-rating on feedback correctness, which was slightly more optimistic than the corresponding human grade.

Qualitative tasks. We additionally ran two qualitative checks. First, we asked a solve-then-comment workflow to proofread all 65 exemplary solutions and the workflow indeed found several errors which we confirmed, with no hallucinations.² Second, we constructed five stress-test problems designed to expose mathematical or contextual failures, including overpowered methods, surprising elementary arguments, advanced material, and multilingual input.

Examples of Feedback generated. We illustrate our feedback generation system with a stress-test example, using the ms-w-example workflow and GPT-5 as the underlying language model. Table 2

²There were false positives, but these were due to the model missing course-specific context.

shows one of our stress tests on which the workflow generates correct and useful feedback. The main difficulty in this example is that when given the problem, the models we tested themselves tend to suggest solutions that have the same shortcomings as the student solution, and will only find elementary arguments if specifically instructed to do so.

3 Results

Grade correlation. Table 3 shows Kendall τ between regraded and human grades. The ms-w-example workflow with GPT-4.1 achieves $\tau = 0.561$, substantially exceeding the human inter-grader baseline from our two independent gradings of the dataset ($\tau = 0.387$). The same workflow consistently outperforms the *baseline_concise* workflow across both models.

When developing the workflows, we optimised for grade correlation under GPT-4o workflows with GPT-4o grading. The data we collected on the influence of model and grader choice on grade correlations are included in the GitHub repository at https://github.com/agohr/llm_proof_feedback.

GPT-5 feedback is substantially better under the HT rubric. Table 4 presents mean HT scores across all four conditions, averaged over both graders. GPT-5 outscores GPT-4.1 on every dimension, most dramatically on correctness (D1: 4.40–4.78 vs. 3.48–4.54) and process guidance (D3: 4.14–4.46 vs. 3.29–3.57). Obviously, the conclusions that can be drawn from this are weaker than conclusions from observed learning outcomes would be.

Self-regulation is hard to grade. Across all conditions and graders, self-regulation support clusters around low values. Indeed, under the GPT-5.4 grader, most items receive a score of exactly 3. We attribute this mainly to the fact that LLMs have difficulty separating this feedback dimension from the others; when manually producing meta-feedback, we found that we faced the same problem.

Cross-grader agreement and bias. Both graders rate GPT-5 feedback substantially higher than GPT-4.1 feedback across all dimensions. The consistency of the GPT-5 advantage across graders lends some support to the validity of the evaluation, as under self-serving bias (Panickssery et al., 2024) each model is expected to rate its own outputs higher.

Qualitative evaluation. We also evaluated our system qualitatively on the two additional tasks defined above.

The first involved proofreading the exemplary solutions to our exercises. We used a solve-then-comment workflow in which the model (o4-mini) first generated a solution to each problem and then evaluated the proposed exemplary solution. Except where the model was unaware of course-specific notation, conventions, or disallowed tools, the feedback was uniformly useful, surfacing several clerical and minor technical errors.

The second was a stress test: five problems that are mathematically accessible but constructed to challenge current LLMs. Performance differed substantially between non-reasoning and reasoning models. While even GPT-5 failed on 3/5 items, targeted hints substantially improved its performance. This suggests that lightweight teacher interventions can meaningfully improve model performance at considerably less effort than writing full question-specific rubrics. Full details are included in the code and data repository.

4 Discussion

Our results support three main findings relevant to automated feedback evaluation and design.

Precomputation of parts of a workflow can help prevent failures. In our stress tests, we were able to trace failure cases to errors in the model-generated mark schemes. Since this stage of feedback generation does not depend on student submissions, it is natural to precompute it at the beginning of a course. Instructors can thus audit system performance on course content before deployment and intervene with suitable hints.

Grade correlation is an insufficient but useful metric for workflow evaluation. On its own, correlation of grades derived from feedback generated by our system is insufficient to select a good workflow for a given application. Our work shows that at least for our undergraduate exercise setting, it fails to show any advantage of the use of reasoning models, when several other metrics we consider, as well as our stress testing, do show such advantage. We still regard grade correlation as a useful screening tool for the development of mathematics tutoring applications, however, as it allows the reuse of graded proof datasets such as (Frieder et al., 2023; Ma et al., 2025).

Question	Show that there are infinitely many numbers that are divisible by 6 and that can be written as the sum of two prime numbers. (Early undergraduate elementary number theory course.)
Student answer	By Dirichlet’s theorem on primes in arithmetic progressions, there are infinitely many primes both of the form $6k + 1$ and $6k + 5$. Let S_5 and S_7 denote two such infinite sets. Then $p + q$ with $p \in S_5$ and $q \in S_7$ is always divisible by 6, and there are infinitely many such pairs; this yields the desired result.
Generated feed-back	<i>What you did well:</i> You correctly identified the relevant residue classes (primes > 3 are $\equiv 1$ or $5 \pmod{6}$), and $1 + 5 \equiv 0 \pmod{6}$), and the overall pairing strategy is appropriate. <i>Main issues:</i> (i) You invoke Dirichlet’s theorem without proof. For this course and problem, relying on such a deep result is not acceptable; an elementary argument is expected. (ii) Even if Dirichlet were allowed, “there are infinitely many such pairs” does not by itself rule out repeated sums—you need infinitely many <i>distinct</i> multiples of 6. <i>How to fix:</i> Replace Dirichlet with an elementary Euclid-style argument: assume finitely many primes $\equiv 5 \pmod{6}$, form $N = 6 \prod r_i - 1$, show $2 \nmid N$ and $3 \nmid N$, and deduce N has a new prime factor $q \equiv 5 \pmod{6}$. Then fix $r = 7 (\equiv 1 \pmod{6})$ and note the sums $q + 7$ are distinct and divisible by 6. <i>Notation:</i> S_5 and S_7 do not match the stated sets of primes of the form $6k + 1$ and $6k + 5$; there is no residue class “7” mod 6.

Table 2: Stress-test example: GPT-5 with the ms-w-example workflow. The student invokes Dirichlet’s theorem. Their argument is mathematically correct, but uses far heavier machinery than the course level allows, and introduces an ambiguous notation (S_5, S_7) mismatched with the stated residue classes. The feedback correctly flags all problems and sketches a possible elementary fix via a Euclid-like argument.

Model	Workflow	τ	n
GPT-4.1	Baseline	+0.306	65
GPT-4.1	MS-w-example	+0.561	65
GPT-5	Baseline	+0.136	65
GPT-5	MS-w-example	+0.296	65
<i>Human inter-grader</i>		+0.387	65

Table 3: Grade correlation: Kendall τ between grades reggraded by GPT-4.1-nano from feedback alone and reconciled human grades. Human baseline shown for reference.

Model	Workflow	D1 Corr.	D2 Task	D3 Proc.	D4 Self-R.
<i>GPT-5.4 grader</i>					
GPT-4.1	Baseline	3.62	3.42	3.29	3.02
GPT-4.1	MS-w-ex.	3.48	3.71	3.40	2.88
GPT-5	Baseline	4.40	4.14	4.14	3.05
GPT-5	MS-w-ex.	4.63	4.29	4.25	3.06
<i>GPT-4.1 grader</i>					
GPT-4.1	Baseline	4.37	3.69	3.57	2.78
GPT-4.1	MS-w-ex.	4.54	3.89	3.46	2.32
GPT-5	Baseline	4.77	4.09	4.28	3.62
GPT-5	MS-w-ex.	4.78	4.25	4.46	3.51

Table 4: Mean HT scores (0–5) by condition and grader. D1–D4 correspond to correctness, task clarity, process guidance, and self-regulation.

Metacognitive feedback is still a gap. In our test cases, our system generally produces correct and task-level relevant feedback but qualitatively, we see that it often fails or does not attempt to diagnose *why* a certain mistake was made. This is also visible to some extent in the self-regulation scores produced by our evaluation, although that part of the evaluation probably suffered from difficulties our meta-grader had in distinguishing this dimension from the others.

We expect that closing this gap would require longer interaction histories and more agentic model scaffolds.

Limitations

Our dataset consists of 65 controlled, author-written items from a single institution and course. This allows useful comparisons to be drawn between configurations, but the frequency and variety of errors in authentic student work may be

different. The HT evaluation is itself performed by LLMs, and only the GPT-4.1/MS-w-example feedback ratings were cross-checked by the authors. We evaluate only two feedback-generation models and two workflows in the main experiment, although our supplementary code and data contain grade-correlation and stress-testing data for additional combinations. Finally, we do not measure learning outcomes or student uptake of the feedback, which would be the gold-standard test for pedagogical effectiveness. While we deployed our tutoring system in several real courses using the lambda feedback platform at Imperial College, and received informal feedback on its use from students, no data from these deployments was used in this paper for ethics reasons. We leave this as future work.

References

- Simon Frieder, Luca Pinchetti, Chevalier Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023. Mathematical Capabilities of ChatGPT. *Advances in neural information processing systems*, 36:27699–27744.
- Abigail S. Gertner and Kurt VanLehn. 2000. Andes: A Coached Problem Solving Environment for Physics. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, pages 133–142.
- John Hattie and Helen Timperley. 2007. [The power of feedback](#). *Review of Educational Research*, 77(1):81–112.
- Irina Jurenka, Markus Kunesch, Kevin R McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, and 1 others. 2024. Towards responsible development of generative AI for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687*.
- Wenjie Ma, Andrei Cojocaru, Neel Kolhe, Bradley Louie, Robin Said Sharif, Haihan Zhang, Vincent Zhuang, Matei Zaharia, and Sewon Min. 2025. Reliable fine-grained evaluation of natural language math proofs. *arXiv preprint arXiv:2510.13888*.
- Pepper Miller and Kristen DiCerbo. 2024. [LLM Based Math Tutoring: Challenges and Dataset](#). Khan Academy.
- Robert C. Moore. 2016. [Mathematics Professors’ Evaluation of Students’ Proofs: A Complex Teaching Practice](#). *International Journal of Research in Undergraduate Mathematics Education*, 2(2):246–278.
- OpenAI. 2024. [Learning to Reason with LLMs](#).
- Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. LLM Evaluators Recognize and Favor Their Own Generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.
- Manooshree Patel, Rayna Bhattacharyya, Thomas Lu, Arnab Mehta, Niels Voss, Narges Norouzi, and Gireeja Ranade. 2026. LeanTutor: Towards a Verified AI Mathematical Proof Tutor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 40670–40678.
- Jelle Wemmenhove, Dick Arends, Thijs Beurskens, Maitreyee Bhaid, Sean McCarren, Jan Moraal, Diego Rivera Garrido, David Tuin, Malcolm Vassallo, Pieter Wils, and 1 others. 2022. Waterproof: Educational Software for Learning How to Write Mathematical Proofs. *arXiv preprint arXiv:2211.13513*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhaghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623.

Confirming Correct, Missing the Rest: LLM Tutoring Agents Struggle Where Feedback Matters Most

Tahreem Yasir Wenbo Li Sam Gilson
 Sutapa Dey Tithi Xiaoyi Tian Tiffany Barnes
 North Carolina State University
 {tyasir, wli55, sagilson, stithi, xtian9, tmbarnes}@ncsu.edu

Abstract

Effective tutoring requires distinguishing optimal, valid but suboptimal, and incorrect student solutions, a distinction central to intelligent tutoring systems (ITS) but untested for LLM-based tutors. As LLMs are increasingly explored as conversational complements to ITS, evaluating their diagnostic precision is essential. We present a benchmark of seven LLM feedback agents in propositional logic using knowledge-graph-derived ground truth across 10,836 solution–feedback pairs and three feedback conditions¹. Models achieved near-ceiling performance on optimal steps but systematically over-rejected valid but suboptimal reasoning and over-validated incorrect solutions, precisely where adaptive tutoring matters most. These failures persisted across models regardless of solution context, suggesting architectural rather than informational limits. Moreover, accurate diagnosis did not reliably produce pedagogically actionable feedback, revealing a gap between diagnostic judgment and instructional effectiveness. Our findings suggest that LLMs are better suited for hybrid architectures where KG-grounded models handle diagnosis while LLMs support open-ended scaffolding and dialogue.

1 Introduction

Effective tutoring requires recognizing not just whether a student solution is correct, but whether it shows good reasoning that should be encouraged or guided toward a more efficient approach. (Gupta et al., 2025). In structured reasoning domains like propositional logic, students may produce optimal next steps that follow expert authored paths to conclusion, apply valid but suboptimal inference rules (*valid-alternative* steps), or make errors entirely (Maniktala et al., 2023). Examples of expert authored (optimal) solution and a valid but longer solution is illustrated in Figure 1.

¹https://github.com/tahreem/BEA_2026

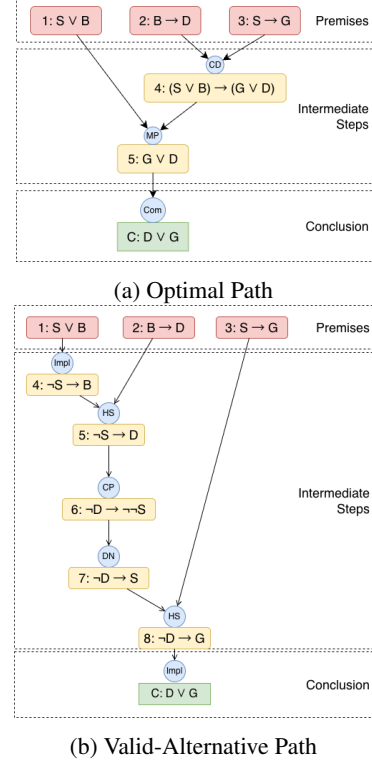


Figure 1: Optimal and valid-alternative solutions (*blue nodes represent abbreviated inference rule names, explained in Table 4*)

Distinguishing among these three types of student steps has direct pedagogical implications identified as *assistance dilemma* by Learning Science experts (Koedinger and Aleven, 2007). Treating valid reasoning as incorrect can discourage productive exploration, while accepting all valid alternative steps without guidance can reinforce inefficient problem solving strategies. Intelligent Tutoring Systems (ITS) address this trade-off through step-level diagnosis of student reasoning using explicitly modeled solution spaces (VanLehn, 2011). However, they rely on expert-authored solution graphs for each problem, limiting scalability to new domains (Weitekamp et al., 2020), and provide limited support for conversational and exploratory dia-

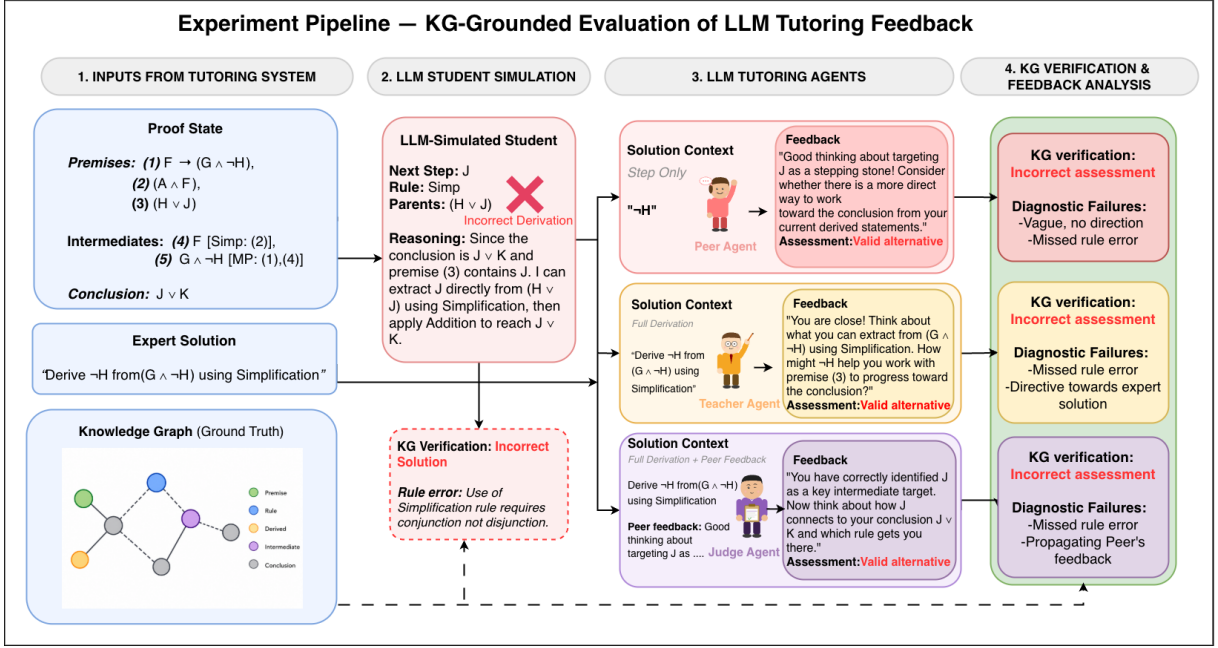


Figure 2: KG-grounded evaluation pipeline. (1) Proof state and expert solution are extracted from tutoring system. (2) LLM-simulated student produces an incorrect derivation (KG verification: Incorrect). (3) Three feedback agents assess student solution as "Valid-Alternative" incorrectly. (4) Automated and manual feedback analysis illustrates systematic over-validation and distinct diagnostic failure modes across Peer, Teacher, and Judge conditions.

logue when students diverge from expected reasoning paths (Zerkouk and Chikhaoui, 2025).

Large language models (LLMs) address these limitations through conversational flexibility, cross-domain generalization, and scalable feedback without per-problem authoring (Reddig et al., 2025; Chen et al., 2025), making them promising complements to ITS. However, without grounding, LLMs hallucinate (Liu et al., 2025a) and often reveal solutions instead of scaffolding reasoning (Macina et al., 2023), limiting their reliability for tutoring. Role-specialized feedback pipelines grounded in expert solutions may improve diagnostic accuracy (Phung et al., 2024; Guo et al., 2024), but whether they support the fine-grained reasoning diagnosis required for effective tutoring remains unclear, especially in structured reasoning domains.

We address this gap through a large-scale evaluation of LLM feedback in propositional logic, where distinguishing between optimal, valid-alternative, and incorrect steps has direct pedagogical importance. Unlike prior tutoring evaluations that rely on binary correctness labels (Gupta et al., 2025; Borchers and Shou, 2025), our framework uses a knowledge-graph (KG) solution space derived from a real tutoring system to represent all valid inference paths (Barnes and Stamper, 2008), enabling grounded three-way diagnosis central to intelligent

tutoring systems (Alevan et al., 2009; VanLehn, 2006).

Contributions: We present the first benchmark for step-level evaluation of LLM tutoring feedback across multiple valid solution paths. Using this benchmark, we evaluate 10,836 LLM-simulated next-step solutions and their feedback with KG-grounded evaluation across seven models and three feedback conditions. To understand when LLM feedback remains pedagogically effective, and what factors influence this behavior, we study three research questions:

– *RQ1:* How accurately LLM feedback agents classify single-step logic proof solutions as optimal, valid-alternative, or incorrect, and what systematic errors emerge?

– *RQ2:* What model- and problem-level factors affect this three-way diagnosis?

– *RQ3:* What is the pedagogical quality of LLM feedback, and how it relates to three-way diagnosis?

Our fine-grained evaluation shows that LLMs consistently over-reject valid-alternative reasoning and over-validate incorrect solutions across models and solution contexts, with failures driven more by model behavior than problem difficulty. We further find that accurate solution diagnosis does not reliably produce pedagogically effective feed-

back. Models may correctly assess solution quality yet still fail to provide guidance that supports learning, reducing feedback to confirmation or error detection manifesting *assistance dilemma*. Together, these findings suggest that current LLMs require ITS-grounded diagnostic mechanisms and are better suited to complement, rather than replace, adaptive tutoring systems.

2 Related Work

2.1 LLM Tutoring and Diagnostic Adaptivity

LLMs show promise for diagnosing student errors (Reddig et al., 2025) and supporting self-regulation (Chen et al., 2025), but tutoring evaluations show they cannot reliably identify reasoning errors, even with reference solutions (Liu et al., 2025b; Jia et al., 2024). Role-specialized feedback pipelines, where one agent generates feedback and another verifies it, can improve precision and reduce over-praise (Phung et al., 2024; Guo et al., 2024), yet verification may also propagate existing errors depending on response type (Guo et al., 2024; Yasir et al., 2026). Prior work further shows that LLMs exhibit limited adaptivity to tutoring context (Borchers and Shou, 2025) and struggle with partially correct responses where guidance is most needed (Mahdavi et al., 2025).

Multiple valid solution paths are common in structured reasoning domains like propositional logic (Große and Renkl, 2006; Maniktala et al., 2023). Traditional ITS address this using manually authored solution graphs that encode both optimal and valid-alternative paths (Aleven et al., 2009), but this limits scalability to pre-authored problems. Whether LLM tutoring agents can similarly perform three-way diagnosis of optimal, valid-alternative, and incorrect responses remains untested and is the central question of this work.

2.2 LLM-Simulated Students

Evaluating tutoring feedback requires student solutions covering optimal, valid-alternative, and incorrect responses, which are often underrepresented in real interaction logs (Maniktala et al., 2023). LLM-simulated students have emerged as a practical solution to this coverage problem. Prior work shows that LLMs can reliably simulate learner actions in open-ended environments (Mannekote et al., 2025), generate responses for evaluating programming tutoring systems (Phung et al., 2024), and produce multiple-choice answers aligned with

real student profiles and responses (Lu and Wang, 2024). LLMs have also been shown to simulate students of different skill levels across educational domains (Benedetto et al., 2024), while generating realistic uncertainty, confusion, and mistakes that effectively challenge teacher agents in benchmarking settings (Shi et al., 2025).

In our work, simulation is additionally required because the tutoring system logs lack reasoning traces, making real interaction data insufficient for step-level reasoning evaluation.

2.3 Evaluation Benchmarks

Existing logic reasoning benchmarks do not support the step-level, multi-path evaluation required in our framework. ProofWriter (Tafjord et al., 2020) evaluates only final proof validity, obscuring step-level rule application. FOLIO (Han et al., 2024) provides expert-authored first-order logic problems but penalizes valid alternative derivations, while ProntoQA (Saparov and He, 2023) supports controlled evaluation with synthetic reasoning chains but cannot represent branching points where multiple inference rules may apply. LogicLearner (Inamdar et al., 2025) focuses on guided practice for basic logic problems, and its benchmark data is unavailable. To the best of our knowledge, none of these benchmarks support evaluating feedback quality for intermediate steps when multiple valid solutions exist.

2.4 Reasoning Difficulty in Logic Proofs

Structural complexity is a known challenge in propositional reasoning (Barnes and Stamper, 2008). Nested connectives increase cognitive load and make it harder to extract propositions needed for further derivation, reducing performance as rule complexity increases (Johnson-Laird and Wason, 1970). In logic proof tutoring, students take more time and derive more unnecessary propositions on steps with nested expressions (Shabrina et al., 2024). Structural nesting is also a strong predictor of LLM error in propositional proof construction, with incorrect steps associated with longer and more complex parent statements across models and prompting conditions (Tithi et al., 2025).

Proof step position introduces an additional challenge. Early steps require reasoning over many valid continuations, while later steps limit the remaining derivations (Shabrina et al., 2024). We therefore examine whether diagnostic failures are

driven by step complexity, proof position, or model-specific factors (Section 4).

3 Dataset and Knowledge Graph

3.1 Task Formulation

A propositional logic proof problem is defined as a tuple $(\mathcal{P}, C)$, where $\mathcal{P} = p_1, \dots, p_n$ is the set of premises (statements assumed to be true) and C is the target conclusion. A proof state $\sigma = (\mathcal{P}, I, C)$ represents progress through the proof, where I is the ordered set of intermediate statements derived by applying inference rules $r \in R$ to statements in $\mathcal{P} \cup I$. Initially, $I = \emptyset$. A complete list of inference rules is provided in Table 4, Appendix A.

In this study, we investigate single-step prediction: deriving the optimal next step (statement) from $\mathcal{P} \cup I$ using exactly one inference rule application that minimizes the remaining derivation distance to C . We restrict the task to single-step prediction and the corresponding feedback generation and evaluation.

3.2 Dataset

To generate single step solution-feedback pairs, we use 516 unique proof states extracted from interaction logs of a propositional logic tutoring system deployed in an undergraduate Discrete Mathematics course at a large U.S. university (Barnes and Stamper, 2008) in Spring 2023. States are drawn from 32 proof problems across five practice levels of varying difficulty (introductory through expert), that provide context specific and on-demand hints we call solution context. Levels 1 (pre-test) and 7 (post-test) are excluded because they do not provide hints. Each instance is unique at triple level $(\mathcal{P}, I, C)$, to ensure a distinct proof state. An example proof state is illustrated in Figure 2 and Figure 7. Table 5, Appendix B summarizes data distribution across tutor difficulty levels, tutoring system illustrations are present in Appendix C.

Step Complexity: To further characterize the reasoning difficulty of proof states in our dataset, we also present step complexity, computed as a weighted sum of operators and nesting structure in the derived expression. Details on computing complexity are available in (Appendix E.1). In our dataset, complexity values ranges from 0 (atomic expressions with no logical operators) to 11 ($M = 3.07, SD = 2.5$), covering simple single-operator next steps through deeply nested compound expressions.

3.3 Knowledge Graph Construction

To provide ground truth that captures all logically valid solution paths in our dataset, we encode the complete solution space for each of the 32 problems as a knowledge graph $K = (\Sigma, E)$. Σ is the set of nodes representing all reachable proof states while E comprise the set of edges representing a valid single step inference between two proof states $(\sigma_i, \sigma_j) \in E$. The proof state σ_j is obtained from σ_i by deriving a new statement $\hat{s}$ via rule $r \in R$ applied to parent statements from $\mathcal{P} \cup I_i$, such that $I_j = I_i \cup \{\hat{s}\}$. The graph is rooted at $\sigma_0 = (\mathcal{P}, \emptyset, C)$ and terminates at σ^* where $C \in I^*$. Given a finite premise set and closed rule set R of 15 propositional inference rules, all reachable derivations are enumerable via forward chaining, making this encoding exhaustive by construction (Barnes and Stamper, 2008).

Each edge $(\sigma_i, \sigma_j) \in E$ in the KG is annotated with the inference rule and respective parent statements in σ_i used to derive the next step $\hat{s}$ in σ_j . This annotation is necessary because a predicted step $\hat{s}$ may be symbolically valid yet inferentially unjustified if attributed to an incorrect rule or unsupported parent statements, illustrated in Figure 2 (*student reasoning error*).

Example: a model predicting C via Modus Ponens must correctly identify both $(A \rightarrow C)$ and A as parents; a prediction that produces C but misattributes the rule or parent statements is flagged as incorrect regardless of symbolic validity. Edge-level annotation thus enables verification of both what was derived and how the derivation is justified.

Distance to conclusion: is defined as the minimum number of derivation steps from the current proof state to the conclusion such that $d(\sigma) = \min_{\pi: \sigma \rightsquigarrow \sigma^*} |\pi|$, and computed via breadth-first search over K . In our dataset it ranges from $[X]$ to $[Y]$ ($M = Z, SD = W$), covering both early proof states with many remaining steps and late proof states close to the conclusion. Unlike prior benchmarks relying on binary correctness, our KG encodes all valid inference paths, enabling three-way diagnosis. A next step, $\hat{s}$ from state σ_t is classified as one of three categories as follows:

Optimal: following an expert-authored path, if $(\sigma_t, \hat{\sigma}_{t+1}) \in E$ and $d(\hat{\sigma}_{t+1}) = d(\sigma_t) - 1$: is a valid inference that strictly reduces distance to the conclusion.

Valid-alternative: if $(\sigma_t, \hat{\sigma}_{t+1}) \in E$ but

$d(\hat{\sigma}_{t+1}) > d(\hat{\sigma}_{t-1})$, that means it does not reduce distance to the conclusion.

Incorrect: $(\sigma_t, \hat{\sigma}_{t+1}) \notin E$, meaning $\hat{\sigma}$ not derivable under any valid inference rule per K .

4 Methods

4.1 Experimental Pipeline

The experimental pipeline extends the framework established in (Yasir et al., 2026) by introducing three-way KG-grounded diagnosis, distinguishing valid-alternative from incorrect solutions, and analyzing over-rejection and over-validation as pedagogically distinct failure modes. We construct next-step solution-feedback pairs using LLM-simulated next-step solutions (Section 4.2) generated from proof states in our dataset (Section 3.2), paired with corresponding LLM feedback under three prompt-level role-specialized feedback conditions (Section 4.3). Figure 1 summarizes this pipeline.

We evaluate seven LLMs, including reasoning-augmented models (GPT-o3, DeepSeek-R1, Qwen-3-32B) and instruction-tuned models (GPT-4.1, Gemini-1.5-Pro, LLaMA-3.3-70B, Mistral-Large). The set includes both proprietary and open-weight models, enabling comparison across capability levels relevant to educational deployment. Each model serves in two roles: (1) as student simulator generating solutions for all 516 proof states and (2) feedback agent evaluating all solutions under the three feedback conditions (Section 4.3). This produces $516 \times 7 \times 3 = 10,836$ solution-feedback pairs (516 proof states, seven LLMs, and three feedback conditions). This design, avoids bias from mismatched model pairs and enable model-level analysis of solution diagnosis and feedback quality across feedback conditions, directly addressing RQ1 and RQ2. Each pair represents one diagnosis instance: a next-step solution and reasoning trace generated by a model and evaluated under one feedback condition. We reuse the same simulated solutions across conditions for each model to ensure fair comparison.

4.2 Student Simulation

The tutoring system logs record only the predicted next step, inference rule, and parent statements, but not the reasoning trace. Without it, evaluating whether LLM feedback agents assess reasoning validity in addition to symbolic correctness is not possible. To address this, we use LLM simulation to generate solutions with reasoning traces. LLM-based simulation is an established methodology

for tutoring evaluation (Mannekote et al., 2025; Phung et al., 2024), and prior work shows that simulated responses contain realistic confusion and mistakes suitable for benchmarking teacher models (Shi et al., 2025).

The student simulator prompt follows the design established in (Yasir et al., 2026). Each model receives a proof state $\sigma = (\mathcal{P}, I, C)$ and generates: (1) a predicted next step in symbolic form, (2) the inference rule used, (3) parent statements, and (4) a natural language reasoning trace. The complete prompt is shown in Figure 14. Models generate the most appropriate next step for the current proof state without instruction to produce a specific solution category. The solution category (optimal, valid-alternative, or incorrect) is later determined by comparison with the KG (Section 3.3). Error patterns observed during human evaluation for response annotation align with common student errors in the tutoring logs, including rule misapplication and incorrect parent selection.

4.3 Feedback Conditions

Accurate diagnosis alone is insufficient for effective tutoring and must be translated into pedagogically appropriate feedback. **Optimal** steps may require only confirmation and encouragement to continue, **Valid-alternative** steps require acknowledging correct reasoning while guiding toward a more efficient approach, and **Incorrect** steps require error identification and corrective guidance (VanLehn, 2006).

To evaluate whether LLM feedback agents can meet these requirements, and whether performance depends on solution context, we define three prompt-level feedback conditions that differ only in the expert solution information provided to the agent, keeping the task and other constraints constant. The three feedback conditions and their prompts are adopted from (Yasir et al., 2026). Each condition represents a distinct tutoring role:

- **Peer** receives only the next step in symbolic form, representing a peer who knows the answer but lacks effective reasoning. This condition tests whether correctness-only grounding is sufficient for accurate diagnosis and pedagogically appropriate feedback.
- **Teacher** represents a real life teacher with complete solution, including the inference rule and parent statements. This tests whether full derivational context improves diagnostic accuracy and feedback quality beyond the Peer condition.

– **Judge** receives the complete solution (inference rule and parent statements) along with the Peer’s feedback, representing a teacher reviewing a teaching assistant’s feedback. This setup is consistent with evaluation and verification loops in ITS (Rodrigues et al., 2025) and aligns with the LLM-as-Judge paradigm (Zheng et al., 2023). This condition tests whether downstream verification can correct upstream diagnostic errors and improve pedagogical feedback quality. Figure 1 shows the information available in each feedback agent.

All feedback agents are prompt-level conditions rather than distinct model architectures. Pedagogical constraints were kept constant across conditions: agents were instructed to scaffold reasoning without revealing answers, acknowledge correct reasoning before addressing errors, and limit feedback to 2–3 sentences (VanLehn, 2006). This design isolates information access as the main independent variable, enabling direct comparison of whether richer solution context improves diagnostic accuracy (RQ1) and feedback quality (RQ3). Prompt templates are provided in Appendix J.

5 Evaluation

5.1 Automated Metrics

To address RQ1 and RQ2, we compute three-way diagnostic performance and identify factors governing misdiagnosis using automated metrics derived from our KG ground truth.

Classification Performance (RQ1) Each feedback agent is prompted to label the solution as Optimal, Valid-alternative, or Incorrect. We compare these labels with KG-derived ground truth to compute F1 scores. However, F1 alone obscures pedagogically distinct failure modes: labeling valid-alternative reasoning as incorrect discourages productive exploration, while labeling incorrect solutions as valid reinforces misconceptions. To capture these two horns of the assistance dilemma, we define two conditional error rates:

Over-rejection (OR): is the rate at which valid-alternative solutions are incorrectly labeled as incorrect, discouraging productive reasoning.

Over-validation (OV): is the rate at which incorrect solutions are labeled as valid, reinforcing misconceptions.

Factors affecting misclassification (RQ2) To identify whether model-level or problem-level reasoning factors affect diagnostic performance, and

whether the effects differ across solution categories, we analyze four factors:

Model vs. solution context: We compare variance (η^2) explained by model selection and feedback condition to determine whether diagnostic accuracy is driven by model capability or solution context.

Step complexity: We test whether step complexity predicts misdiagnosis to separate reasoning difficulty from model-level diagnostic failures. Nested logical structures are a known source of reasoning difficulty for both humans (Johnson-Laird and Wason, 1970; Shabrina et al., 2024) and LLMs (Tithi et al., 2025). Full computation details are in Appendix E.1.

Distance to goal: We test whether distance to goal (Section 3.3) predicts OR and OV at different derivation stages. Together with step complexity, this helps determine whether failures reflect problem difficulty or model-level biases.

Inference rule: We compare F1 scores across inference rules to identify rules systematically misdiagnosed across models.

Table 1: Rubric for evaluating pedagogical quality of agent-generated feedback.

Dimension	Score 1: Low	Score 2: Moderate	Score 3: High
Correctness	Contains errors that would mislead the student	Minor imprecisions but no fundamental errors	All statements are logically and factually accurate
Error Identification	Fails to identify the error or identifies incorrectly	Indicates error exists but identification is vague	Precisely identifies the specific error; affirms correctness if applicable
Revealing	Guides without revealing, using scaffolding or Socratic questioning	Hints at solution direction without explicit disclosure	Discloses solution content explicitly (names rule, states next step)
Actionability	No actionable guidance; student cannot proceed	General guidance but lacks specificity	Clear, specific guidance enabling appropriate next step

5.2 Human Evaluation (RQ3)

To determine whether better solution diagnosis leads to pedagogically appropriate feedback, two annotators independently evaluated feedback qual-

ity on 100 solution-feedback pairs per condition. We use four rubric dimensions adapted from prior work (Maurya et al., 2025), shown in Table 1. These dimensions directly operationalize the distinction between solution diagnosis and pedagogical feedback quality central to RQ3. Complete details on sample and annotator selection, annotation procedures (calibration and inter-rater reliability), and instructions are provided in Appendix I.

6 Results

We organize results around three research questions. First, we report solution classification performance and systematic errors, i.e., OR and OV (RQ1). Next, we examine how model selection, feedback condition, step complexity, distance to goal, and inference rule affect classification accuracy (RQ2). Finally, we assess whether accurate solution classification translates into pedagogically appropriate feedback (RQ3).

6.1 RQ1: Classification Performance and Error Patterns

Models showed distinct classification patterns across student solutions. Table 2 presents classification performance (F1) across models and feedback conditions. All models achieved near-ceiling performance on optimal student solutions ($F1 : 94 - 99\%$) but struggled with valid-alternatives ($F1 : 0 - 76\%$) and incorrect solutions ($F1 : 4 - 55\%$). GPT-4.1 and GPT-o3 achieved balanced but moderate performance on valid-alternative solutions. Gemini and DeepSeek achieved the highest valid-alternative scores ($F1 : 70-76\%$) but poor incorrect detection ($F1 : 4-19\%$). In contrast, LLaMA 3 showed near-zero valid-alternative diagnosis ($F1 : 0-17\%$) but the highest incorrect detection ($F1 : 45-55\%$).

Table 2 further quantifies these patterns using conditional error probabilities, OR and OV. Gemini and DeepSeek showed high over-validation (69-71%) but low over-rejection (13-17%), treating most student solutions as valid-alternative regardless of correctness. LLaMA 3 showed the opposite pattern: 91% over-rejection but only 6% over-validation, rejecting nearly all non-optimal solutions. GPT-4.1 and GPT-o3 showed moderate rates of both errors (29-41%).

6.2 RQ2: Factors Affecting Classification

Given the substantial variation in classification performance across models, we examined the factors

Table 2: Classification F1 and error rates by model. P = Peer, T = Teacher, J = Judge. OR = Over-Rejection, OV = Over-Validation. CIs: $\pm 5-8\%$. $^\dagger N < 50$.

Model	Opt.	Valid-Alt.			Incorrect			OR	OV
	Avg	P	T	J	P	T	J	(%)	(%)
GPT-4.1	0.99	0.55	0.57	0.49	0.42	0.40	0.41	38.57	28.65
GPT-o3	0.99	0.48	0.55	0.45	0.38	0.40	0.40	40.50	29.97
Gemini-1.5 Pro	0.99	0.70	0.73	0.72	0.09	0.04	0.08	12.77	70.55
DeepSeek	0.99	0.74	0.76	0.74	0.09	0.12	0.19	17.12	68.60
Qwen	0.99	0.58	0.42	0.54	0.11	0.07	0.12	54.65	28.30
Mistral	0.95	0.58	0.47	0.52	0.15	0.09	0.12	26.28	54.37
Llama 3 [†]	0.94	0.03	0.00	0.17	0.48	0.55	0.45	91.07	6.47

Table 3: Mean complexity (correct vs. incorrect steps)

Type	Corr.	Incorr.	Diff.	N
Optimal	1.95	2.46	+0.51*	1189
Valid Alt.	3.62	2.97	-0.65	816
Incorrect	3.91	3.78	-0.13	1619

* $p < .05$ (Mann-Whitney U)

affecting misdiagnosis.

– Model Selection vs. Feedback Conditions:

Model selection explained most of the variance in classification performance ($\eta^2 > 0.95$, $p < .001$), while feedback conditions had negligible effect ($\eta^2 < .01$). Table 2 shows that additional information access did not improve classification when student solutions diverged from the expected path (valid-alternative or incorrect solutions).

– **Step Complexity:** Because feedback conditions had negligible effect on diagnosis ($\eta^2 < .01$), we pooled feedback agents across models to maximize statistical power for complexity analysis. However, complexity predicted misdiagnosis only for Optimal solutions: more complex steps were misdiagnosed more often ($M = 2.46$) than correctly classified ones ($M = 1.95$; $p < .05$).

For valid-alternative and incorrect solutions, complexity did not predict errors ($p > .08$). Feedback agents rejected simpler valid-alternative solutions ($M = 2.97$) more often than complex ones ($M = 3.62$), suggesting rejection was driven by deviation from the expert path rather than step complexity. Step complexity explained negligible variance in diagnosis performance ($\eta^2 < .01$, $p > .05$) compared to model selection ($\eta^2 > .95$, $p < .001$), confirming that failures are model-level rather than difficulty-driven.

To analyze complexity effects on OR and OV, proof states were grouped into low (1-2), medium (3-4), and high (5+) complexity levels, approximately corresponding to the 33rd and 66th percentile of complexity distribution ($P_{33} = 2$, $P_{66} =$

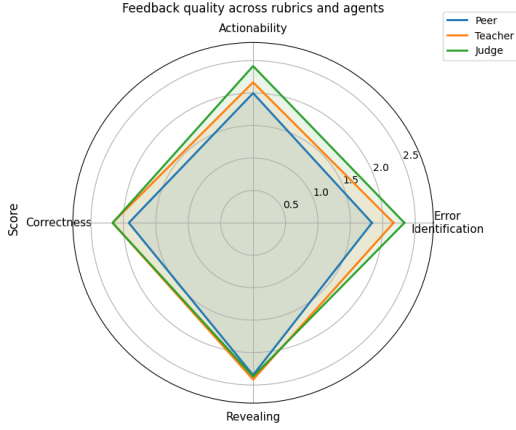


Figure 3: Mean feedback quality scores across rubric dimensions (N=100).

3). Figure 18 shows that LLaMA over-rejection approaches 100% at higher complexity tiers, while DeepSeek and Gemini over-validation persists across all tiers, further supporting model-level rather than difficulty-driven failure.

– **Distance to Conclusion:** Distance tiers used the same percentile-based partition: near ($d \leq 2$), mid ($2 < d \leq 3$), and far ($d > 3$). Figure 19 shows that over-rejection is highest at near distances, meaning agents most often penalize valid-alternative reasoning when solutions are closest to the conclusion. Distance to conclusion explained negligible variance ($\eta^2 < .01$), again supporting model-level rather than position-driven failures.

– **Inference Rules:** Analysis of inference rules and F1 scores (Figure 20) shows that failures concentrate on structurally complex rules (MT, CD, DeM). LLaMA showed minimal gains across most rule types, while Gemini and DeepSeek consistently achieved the highest scores.

6.3 RQ3: Feedback Quality

To assess whether accurate solution classification translates into pedagogically appropriate feedback, we report aggregate scores and LLM responses.

Overall patterns: Figure 3 summarizes mean rubric scores (1–3 scale). Error Identification here measures whether feedback precisely identified the reasoning flaw, distinct from classification performance (RQ1), which captures only categorical judgment. Across agents, scores were lowest on Error Identification and Actionability, indicating that feedback frequently failed to diagnose errors or provide clear guidance, consistent with automated analysis (Section 6.1 and 6.2).

Qualitative analysis further explains this pat-

tern. Although prompts instructed agents to acknowledge correct reasoning *before addressing errors*, agents often praised solutions without identifying errors or giving actionable guidance. This produced encouraging but pedagogically misleading feedback, especially for incorrect and valid-alternative solutions. (Example 1 and 2 below)

Peer feedback: Peer agents consistently received the lowest scores across all human-evaluated dimensions (Figure 3). Limited solution context led Peer agents to restate student incorrect reasoning or provide vague feedback without diagnosing errors or offering guidance (Example 1 below and Figure 8 in Appendix H).

Example 1: Vague Peer feedback (repeats student’s mistakes)
Proof state: 1. $((S \rightarrow Y) \vee (I \wedge Q))$, 2. $((I \wedge Q) \rightarrow D)$, 3. $\neg D$, 4. $((S \rightarrow Y) \rightarrow D)$, 5. $\neg(S \rightarrow Y)$, 6. $(I \wedge Q)$
Solution context: D
Student reasoning: *Use Modus Tollens on $((S \rightarrow Y) \rightarrow D)$ and $\neg D$...*
Student error: Rule cant be applied on parent statements
Peer Feedback: *"Great job on identifying Modus Tollens on $((S \rightarrow Y) \rightarrow D)$ and $\neg D$ as a key step. Now ask yourself, is there a step that could bridge your current progress to the CONCLUSION more efficiently..."*

Teacher and Judge feedback: Teacher and Judge agents performed better at error identification than Peer agents. However, their feedback often penalized valid-alternative reasoning by redirecting students toward the provided solution context. This pattern was strongest for complex inference rules, which were also associated with higher error rates in Section 6.2 (Figure 20). In contrast, agents more readily acknowledged valid-alternative reasoning for simpler inference rules (Example 2, 3 and Figure 9).

Example 2: Teacher feedback validating incorrect reasoning
Proof state: 1. $(B \leftrightarrow \neg J)$, 2. $(\neg N \vee J)$, 3. $(B \vee \neg N)$, 4. $((B \rightarrow \neg J) \wedge (\neg J \rightarrow B))$, 5. $(N \rightarrow J)$, 6. $(\neg B \rightarrow \neg N)$
Solution context: *Derive $(J \vee \neg N)$ from $(\neg N \vee J)$ using Commutative rule.*
Student reasoning: *$(B \rightarrow \neg N)$ is optimal because it is the exact form of the conclusion, it can be derived from the contrapositive of $(\neg B \rightarrow \neg N)$...*
Student error: Incorrect derivation
Teacher Feedback: *"Great job recognizing that $(B \rightarrow \neg N)$ is the contrapositive of $(\neg B \rightarrow \neg N)$ and using that to reach the conclusion..."*

Verification Condition Contrary to prior LLM-as-Judge findings (Zheng et al., 2023), Judge feedback did not consistently improve feedback quality. When Peer agents missed errors, Judge agents often propagated rather than corrected them (Figure 12, anchoring Judge feedback to the Peer’s framing inheriting the or exacerbating mistakes. This

asymmetric pattern, where verification does not consistently improve some response is consistent with our prior work (Yasir et al., 2026).

Example 3: Judge repeats Peers’ mistakes and forces towards the available solution context
Proof state: 1. $\neg(\neg G \wedge B)$, 2. $(G \rightarrow D)$
Student reasoning: Applying DN to $\neg(\neg G \wedge B)$ to get $(G \wedge B)$ is optimal because it directly gives us both G and B as available conjuncts...
Student error: Incorrect rule application
Peer Feedback: "Great job applying double negation here! As you move forward, consider how you can use the new conjuncts to get closer..."
Solution context: Derive $(\neg\neg G \vee \neg B)$ from $\neg(\neg G \wedge B)$ using DeM Law.
Judge Feedback: "You made a good attempt by considering double negation, but before applying DN, try using De Morgan’s Law to rewrite..."

7 Discussion

RQ1: Consistent with the high OR and OV rates reported in Section 6.1, LLMs showed limited sensitivity to valid-alternative reasoning, relying on surface alignment with the provided solution context rather than reasoning validity. These failures have direct pedagogical consequences: OR discourages productive exploration and penalizes sound reasoning that diverges from the expert path (Manik-tala et al., 2023), while OV reinforces misconceptions and limits corrective feedback essential for learning (VanLehn, 2006). These failure modes are obscured under binary correctness schemes and only separable through our three-way KG-grounded framework, highlighting the importance of evaluation methodology alongside model capability (VanLehn, 2006; Koedinger et al., 2012).

RQ2: As shown in Section 6.2, model selection explained substantially more variance than feedback condition, indicating that diagnostic failures are primarily model-level rather than informational. Models performed well when solutions aligned with the optimal path but showed systematic bias once reasoning diverged, regardless of the provided solution context. Together with the negligible effects of step complexity, distance to conclusion, and inference rule type, these findings suggest persistent limits in LLM reasoning for constrained domains such as logic proofs. Improving diagnosis therefore likely requires capability-level advances rather than prompt or information-access design (Stamper et al., 2024; Venugopalan et al., 2025).

RQ3: Section 6.3 further showed that accurate diagnosis does not reliably translate into pedagogically effective feedback. Agents with more solution context produced overly directive feedback,

while less-informed agents generated open-ended responses lacking diagnostic grounding. This reflects the assistance dilemma: excessive directness constrains learning, while insufficient guidance fails to support progress (Koedinger and Aleven, 2007). Prompt-level layering alone therefore cannot ensure adaptive feedback. Effective verification likely requires independent diagnostic signals, similar to the grounded student modeling mechanisms used in ITS. Our findings further suggest that verification may require distinct models or explicit reasoning chains enforcing independent evaluation rather than sequential refinement of flawed judgments.

LLMs address ITS limitations in conversational flexibility and cross-domain scalability (Reddig et al., 2025); however, their inability to reliably distinguish valid-alternative from incorrect reasoning risks pedagogically misleading feedback, precisely the problem ITS student modeling is designed to address (Venugopalan et al., 2025). These findings confirm that linguistic fluency alone does not ensure pedagogical effectiveness (Shetye, 2024). Rather than replacing ITS, a more effective hybrid architecture may use LLMs for open-ended dialogue and hint generation while KG-grounded classifiers provide diagnostic labeling, with feedback conditioned on classifier outputs instead of self-assessed reasoning validity.

8 Conclusion

This study benchmarked LLMs as step-level tutors in propositional logic, evaluating both diagnostic accuracy and feedback quality across conditions that vary in solution information access. Our results show that LLMs reliably confirmed optimal steps but struggled with valid-alternative and incorrect solutions, precisely where adaptive tutoring is most critical. Classification failures on valid-alternative and incorrect solutions persisted across all models regardless of solution context, with model selection governing the magnitude of errors. Moreover, accurate classification did not reliably translate into pedagogically actionable feedback. These findings indicate that current LLMs cannot resolve the assistance dilemma without ITS-grounded diagnostic mechanisms. Effective LLM-based tutoring in structured reasoning domains therefore requires hybrid architectures that delegate diagnostic classification to KG-grounded models while leveraging LLMs for open-ended scaffolding and dialogue.

Limitations

This study targets propositional logic by design: KG-grounded exhaustive enumeration of valid inference paths is feasible in this formal domain and provides the principled ground truth that evaluation requires. Findings may not directly generalize to domains where exhaustive solution space enumeration is not tractable or where symbolic constraints are weaker.

Our 516 proof states are drawn from a single undergraduate Discrete Mathematics course at one institution across five difficulty levels, which limits the diversity of reasoning contexts and problem types represented.

Additionally, while LLM-simulated solutions enable controlled evaluation across all three solution categories, they may not fully capture the diversity of authentic student reasoning; planned extensions will investigate alignment between simulated and real student solution patterns using interaction logs.

Feedback agents were evaluated using zero-shot prompting after iterative prompt refinement and we did not see significant performance difference with few-shot prompting while the initial prompt calibration. However, we expect fine-tuning may improve classification performance, particularly for valid-alternative and incorrect categories, and remains a direction for future work.

This study diagnoses systematic classification failures without proposing a mitigation mechanism. The empirical findings are intended to inform the design of hybrid architectures that combine LLM flexibility with KG-grounded diagnostic precision; implementing and evaluating such systems is left to future work.

Our 10,836 solution-feedback pairs capture single-step feedback evaluation and do not model multi-turn student-tutor interactions or cumulative learning gains over time, which are ultimately what adaptive tutoring aims to improve.

Finally, our human evaluation covers 100 solution-feedback pairs per condition across four rubric dimensions; systematic fine-grained analysis of cases where models correctly classify a solution yet still produce inaccurate, vague, revealing, or unactionable feedback would require annotation at a scale, and represents an open methodological challenge for the field.

Ethics Statement

Student interaction data was collected under IRB approval with informed consent and anonymized before analysis. Two annotators with domain expertise in the tutoring system served as expert raters; both are co-authors of this study. To mitigate potential conflict of interest, annotators coded independently with no access to each other’s labels, research hypotheses, or study outcomes during annotation, and rubric definitions were finalized before coding began.

Reproducibility

Code and data can be found here ². All experiments use temperature 0.0 to ensure deterministic, reproducible results, prioritizing controlled comparison over response diversity. While real-world tutoring systems might benefit from non-zero temperature, our evaluation focuses on measuring the systematic effects of multi-agent architectures under controlled conditions. Model specifications and prompts appear in Appendices G and J.

AI Usage Disclosure

This work studies and evaluates large language models as research objects. We utilized large language models as assistive tools during manuscript preparation, including formatting guidelines and brainstorming organization, mainly, and paraphrasing on a need basis only. We did not use any AI tools for designing, implementing, or executing this research study. All the claims, analyses, hypotheses, and conclusions are developed, verified, and reviewed by the authors. Moreover, no AI tool was used for generating or labeling data, making judgments about data, or making any scientific claims. All implementations were reviewed, validated, and finalized by the authors. The authors take full responsibility for the correctness, originality, and integrity of the work.

Acknowledgment

This research was partially supported by the NSF Grant: Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502).

²https://github.com/tahreemm/BEA_2026

References

- Vincent Aleven, Bruce McLaren, Jonathan Sewall, and Kenneth Koedinger. 2009. [A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors](#). *I. J. Artificial Intelligence in Education*, 19:105–154.
- Tiffany Barnes and John Stamper. 2008. [Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data](#). pages 373–382.
- Luca Benedetto, Giovanni Aradelli, Antonia Donvito, Alberto Lucchetti, Andrea Cappelli, and Paula Buttery. 2024. [Using LLMs to simulate students’ responses to exam questions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11351–11368, Miami, Florida, USA. Association for Computational Linguistics.
- Conrad Borchers and Tianze Shou. 2025. Can large language models match tutoring system adaptivity? a benchmarking study. In *International Conference on Artificial Intelligence in Education*, pages 407–420. Springer.
- Liuqiao Chen, Yan Huang, Zhenglin Jiang, and Ying Chen. 2025. Generative Feedback for Code Learning: A Study on LLM-Driven Formative Assessment. In *Proceedings of the 2025 International Conference on AI-enabled Education, AIEE ’25*, pages 253–259, New York, NY, USA. Association for Computing Machinery.
- Cornelia Große and Alexander Renkl. 2006. [Effects of multiple solution methods mathematics learning](#). *Learning and Instruction*, 16(2), 122-138. *Learning and Instruction - LEARN INSTR*, 16:122–138.
- Shuchen Guo, Ehsan Latif, Yifan Zhou, Xuan Huang, and Xiaoming Zhai. 2024. [Using Generative AI and Multi-Agents to Provide Automatic Feedback](#). *arXiv preprint*. ArXiv:2411.07407 [cs].
- Adit Gupta, Jennifer Reddig, Tommaso Calo, Daniel Weitekamp, and Christopher J MacLellan. 2025. Beyond final answers: Evaluating large language models for math tutoring. In *International Conference on Artificial Intelligence in Education*, pages 323–337. Springer.
- Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhen-ting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alex Wardle-Solano, Hannah Szabo, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. [FOLIO: Natural Language Reasoning with First-Order Logic](#). *arXiv preprint*. ArXiv:2209.00840 [cs].
- Amogh Inamdar, Uzay Macar, Michel Vazirani, Michael Tarnow, Zarina Mustapha, Natalia Dittren, Sam Sadeh, Nakul Verma, and Ansaf Salleb-Aouissi. 2025. Logiclearner: A tool for the guided practice of propositional logic proofs. *arXiv preprint arXiv:2503.19280*.
- Qinjin Jia, Jialin Cui, Ruijie Xi, Chengyuan Liu, Parvez Rashid, Ruochi Li, and Edward Gehringer. 2024. On assessing the faithfulness of llm-generated feedback on student assignments. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 491–499.
- PN Johnson-Laird and PC Wason. 1970. Insight into a logical relation. *The Quarterly Journal of Experimental Psychology*, 22(1):49–61.
- Kenneth R Koedinger and Vincent Aleven. 2007. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational psychology review*, 19(3):239–264.
- Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2012. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798.
- Rongxin Liu, Julianna Zhao, Benjamin Xu, Christopher Perez, Yuliia Zhukovets, and David Malan. 2025a. [Improving AI in CS50: Leveraging Human Feedback for Better Learning](#). pages 715–721.
- Rongxin Liu, Julianna Zhao, Benjamin Xu, Christopher Perez, Yuliia Zhukovets, and David J Malan. 2025b. Improving ai in cs50: Leveraging human feedback for better learning. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 715–721.
- Xinyi Lu and Xu Wang. 2024. [Generative students: Using LLM-simulated student profiles to support question item evaluation](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale (L@S ’24)*, pages 16–27, Atlanta, GA, USA. ACM.
- Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.
- Hamed Mahdavi, Pouria Mahdavinia, Samira Malek, Pegah Mohammadipour, Alireza Hashemi, Majid Daliri, Alireza Farhadi, Amir Khasahmadi, Niloo-far Mireshghallah, and Vasant Honavar. 2025. [Ref-Grader: Automated Grading of Mathematical Competition Proofs using Agentic Workflows](#). *arXiv preprint*. ArXiv:2510.09021 [cs].
- Mehak Maniktala, Min Chi, and Tiffany Barnes. 2023. Enhancing a student productivity model for adaptive problem-solving assistance. *User Modeling and User-Adapted Interaction*, 33(1):159–188.
- Amogh Mannekote, Adam Davies, Jina Kang, and Kristy Elizabeth Boyer. 2025. Can llms reliably simulate human learner actions? a simulation authoring framework for open-ended learning environments. In

- Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29044–29052.
- Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI Tutor Evaluation: An Evaluation Taxonomy for Pedagogical Ability Assessment of LLM-Powered AI Tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.
- Tung Phung, Victor-Alexandru Pădurean, Anjali Singh, Christopher Brooks, José Cambronero, Sumit Gulwani, Adish Singla, and Gustavo Soares. 2024. Automating Human Tutor-Style Programming Feedback: Leveraging GPT-4 Tutor Model for Hint Generation and GPT-3.5 Student Model for Hint Validation. In *Proceedings of the 14th Learning Analytics and Knowledge Conference, LAK '24*, pages 12–23, New York, NY, USA. Association for Computing Machinery.
- Jennifer M. Reddig, Arav Arora, and Christopher J. MacLellan. 2025. Generating In-Context, Personalized Feedback for Intelligent Tutors with Large Language Models. *International Journal of Artificial Intelligence in Education*, 35(6):3459–3500.
- Bárbara Rodrigues, Rui Pinto, and Gil Gonçalves. 2025. A systematic literature review of ai-driven intelligent tutoring systems in engineering education: Emphasizing personalization, feedback, and student monitoring. *IEEE Access*, 13:190152–190177.
- Abulhair Saparov and He He. 2023. [Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought](#). *arXiv preprint*. ArXiv:2210.01240 [cs].
- Iulian Vlad Serban and 1 others. 2020. Automated personalized feedback improves learning gains in an intelligent tutoring system. In *International Conference on Artificial Intelligence in Education*, pages 140–146. Springer. Personalized hints improve learning gains; feedback from ZPD.
- Preya Shabrina, Behrooz Mostafavi, Mark Abdelshieed, Min Chi, and Tiffany Barnes. 2024. Investigating the impact of backward strategy learning in a logic tutor: Aiding subgoal learning towards improved problem solving. *International Journal of Artificial Intelligence in Education*, 34(3):825–861.
- Shamini Shetye. 2024. An evaluation of khanmigo, a generative ai tool, as a computer-assisted language learning app. *Studies in Applied Linguistics and TESOL*, 24(1).
- Yao Shi, Rongkeng Liang, and Yong Xu. 2025. Educationq: Evaluating llms’ teaching capabilities through multi-agent dialogue framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32799–32828.
- John Stamper, Ruiwei Xiao, and Xinying Hou. 2024. Enhancing llm-based feedback: Insights from intelligent tutoring systems and the learning sciences. In *International Conference on Artificial Intelligence in Education*, pages 32–43. Springer.
- Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. [Proofwriter: Generating implications, proofs, and abductive statements over natural language](#). *arXiv preprint arXiv:2012.13048*.
- Sutapa Dey Tithi, Arun Kumar Ramesh, Clara Di-Marco, Xiaoyi Tian, Nazia Alam, Kimia Fazeli, and Tiffany Barnes. 2025. [The promise and limits of LLMs in constructing proofs and hints for logic problems in intelligent tutoring systems](#). *arXiv preprint*. ArXiv:2505.04736 [cs].
- Kurt VanLehn. 2006. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265.
- Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221. Step-based ITS achieve effect sizes of 0.75-0.80, comparable to human tutoring (0.79).
- Devika Venugopalan, Ziwen Yan, Conrad Borchers, Jionghao Lin, and Vincent Aleven. 2025. Combining large language models with tutoring system intelligence: A case study in caregiver homework support. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 373–383.
- Daniel Weitekamp, Erik Harpstead, and Ken R. Koedinger. 2020. An Interaction Design for Machine Teaching to Develop AI Tutors. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20*, pages 1–11, New York, NY, USA. Association for Computing Machinery.
- Tahreem Yasir, Sutapa Dey Tithi, Benyamin Tabarsi, Dmitri Droujkov, Sam Gilson Yasitha Rajapaksha, Xiaoyi Tian, Arun Ramesh, Tiffany Barnes, and 1 others. 2026. When verification hurts: Asymmetric effects of multi-agent feedback in logic proof tutoring. *arXiv preprint arXiv:2603.27076*.
- Mohamed Zerkouk and Belkacem Chikhaoui. 2025. A comprehensive review of ai-based intelligent tutoring systems: Applications and challenges. *arXiv preprint arXiv:2507.18882*. ITS can improve student performance; CLT informs stepwise hint design.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Inference Rule List

We employ a fixed set of propositional inference rules used in Logic Tutor for the dataset. The short names were used for consistent response generation and evaluation. The complete list of inference rules, along with short names and derivations are provided below in Table 4.

Table 4: Propositional inference rules used in this work.

Abbrev.	Rule Name	Form
MP	Modus Ponens	$P \rightarrow Q, P \Rightarrow Q$
MT	Modus Tollens	$P \rightarrow Q, \neg Q \Rightarrow \neg P$
Conj	Conjunction	$P, Q \Rightarrow P \wedge Q$
Simp	Simplification	$P \wedge Q \Rightarrow P \text{ (or } Q)$
Add	Addition	$P \Rightarrow P \vee Q$
DS	Disjunctive Syllogism	$P \vee Q, \neg P \Rightarrow Q$
HS	Hypothetical Syllogism	$P \rightarrow Q, Q \rightarrow R \Rightarrow P \rightarrow R$
Impl	Implication	$P \rightarrow Q \equiv \neg P \vee Q$
DN	Double Negation	$P \equiv \neg \neg P$
CP	Contraposition	$P \rightarrow Q \equiv \neg Q \rightarrow \neg P$
Com	Commutation	$P \vee Q \equiv Q \vee P$
Assoc	Associativity	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$
Dist	Distribution	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
CD	Constructive Dilemma	$(P \rightarrow Q), (R \rightarrow S), P \vee R \Rightarrow Q \vee S$
Equiv	Equivalence	$P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

B Dataset Distribution

Table 5: Dataset distribution across practice levels (2-6)

Level	Proof States	Avg. Statements
2 (Introductory)	48	5.86
3 (Basic)	111	7.73
4 (Intermediate)	148	7.94
5 (Advanced)	85	5.64
6 (Expert)	95	6.41
Total	516	6.72

* Levels 1 and 7 excluded because of unavailability of hints.

C Illustrative Logic Tutor Proof Interaction

Figure 4–6 illustrates a representative student interaction in the propositional logic tutor. These screenshots demonstrate forward chaining, rule application, and goal completion within the tutor interface.

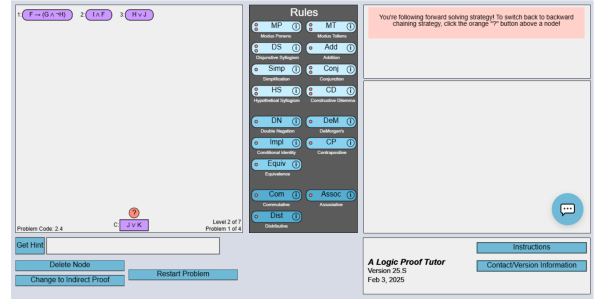


Figure 4: **Initial proof state and goal specification.** The student is presented with the premises (top left) and the target conclusion ($J \vee K$) at the bottom. Available inference rules are displayed on the middle. At this stage, no intermediate steps have been derived, and the student must choose a productive forward step toward the goal.

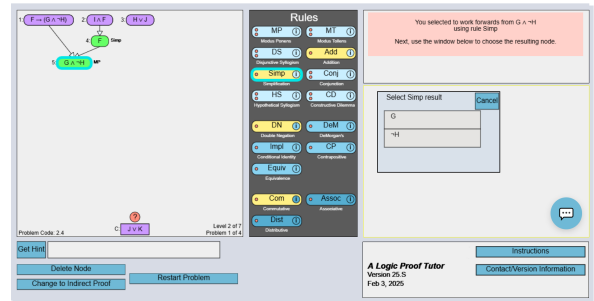


Figure 5: **Rule application with guided simplification.** After deriving an intermediate conjunction via Modus Ponens, the student applies the *Simplification* rule. The interface prompts the learner to select the appropriate resulting literal (G or $\neg H$), illustrating fine-grained, step-level decision making supported by rule constraints.

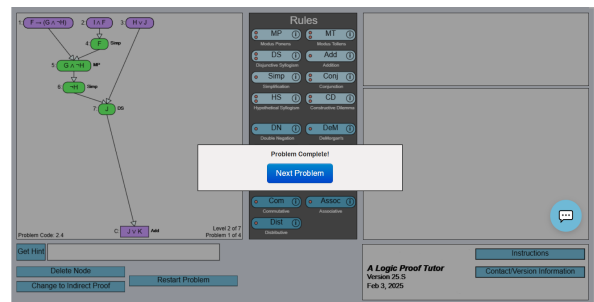


Figure 6: **Successful proof completion.** The student derives J via Disjunctive Syllogism and applies the *Addition* rule to reach the target conclusion $J \vee K$. The system confirms completion, reinforcing correct rule sequencing and alignment with the goal state.

D Representative Proof State

This appendix presents an illustrative propositional logic proof instance Figure 7 annotated with inter-

Representative proof Instance	
Givens	
• (1) $((S \rightarrow Y) \vee (I * Q))$	
• (2) $((I \wedge Q) \rightarrow D)$	
• (3) $\neg D$	
• (4) $((S \rightarrow Y) \rightarrow D)$	
Intermediate Steps	
• (5) $\neg(S \rightarrow Y)$	[MT: (3), (4)]
• (6) $\neg(I \wedge Q)$	[MT: (3), (2)]
• (7) $(S \rightarrow Y)$	[DS: (6), (1)]
Conclusion: Y	
Peer Solution Context	
$\neg(\neg S \vee Y)$	
Teacher & Judge Solution Context	
Derive $\neg(\neg S \vee Y)$ from $\neg(S \rightarrow Y)$ using the Implication rule.	

Figure 7: Illustrative proof instance showing intermediate derivations and differential hint access for Peer and Teacher agents.

mediate derivations and agent-specific hints. Intermediate steps are derived by applying valid inference rules to previously established statements, with each step indexed and linked to its parent statements and rule application. This explicit structure exposes the shortest derivation path toward the conclusion while preserving alternative valid reasoning trajectories.

The Peer is provided only with the KG-verified optimal next symbolic step, enabling guidance that focuses on local progression without access to derivational context. In contrast, the Teacher additionally receives the rule and parent statements used to derive this step, allowing feedback to reference structural reasoning without revealing the answer directly. This distinction isolates the effect of solution access: the Peer operates under minimal grounding, while the Teacher leverages full derivational context to support more informed intervention. The representative proof state is available in Figure 7.

E Knowledge Graph based Evaluation Metrics

This appendix provides formal definitions and illustrative examples of the graph-based metrics used to evaluate next-step reasoning quality. All metrics are computed with respect to the knowledge graph (KG) constructed for each problem.

E.1 Step Complexity

Step complexity is computed as a nesting-weighted operator count:

$$c(\phi) = \sum_{o \in \mathcal{O}(\phi)} w(o, \phi)$$

where $w(o, \phi) = w_{\text{base}}(o) + w_{\text{nest}}(o)$ if o applies to a parenthesized subexpression in ϕ , and $w(o, \phi) = w_{\text{base}}(o)$ otherwise. Each opening parenthesis contributes an additional unit to capture structural depth.

Higher-order connectives receive larger nesting penalties because applying them to compound arguments requires tracking negation across multiple sub-propositions simultaneously (Johnson-Laird and Wason, 1970; Tithi et al., 2025). Atomic expressions containing no operators receive a score of 0.

For example: $c(F) = 0$; $c(A \vee B) = 1$; $c(A \rightarrow (B \vee C)) = 6$, where the implication receives its base weight plus a nesting penalty for applying to the parenthesized subexpression $(B \vee C)$.

F Implementation Details

Response Constraints. We standardize inference rule references using abbreviations (MP, MT, DS, HS, DeM, Impl) for consistent evaluation. The complete list appears in Table 4.

Token Limits. Rather than hard limits, we employ soft constraints via prompts (2–3 sentences).

Quality Validation. Two automated mechanisms ensure data integrity: (1) zero-token outputs trigger automatic retry, and (2) responses missing required JSON fields or violating constraints trigger re-prompting. Persistent failures after three attempts are flagged for manual review. Across 516 instances on 7 models, fewer than 3% required retry.

G Model Specifications

Please refer to Table 6.

Table 6: Models used in our experiments

Model	Org.	API	Context	Param
GPT-4.1	OpenAI	OpenAI	1M	-
GPT-o3	OpenAI	OpenAI	1M	-
Gemini 1.5 Pro	Google	Vertex	128K	-
Qwen-3-32B	Alibaba	Groq	32K	32B
LLaMA-3.3-70B	Meta	Groq	128K	70B
Mistral Large	Mistral	Mistral	256K	675B
Deepseek-r1-0528	Deepseek	Deepinfra	128K	671B

"-" refers to undisclosed

H Representative Examples of Failure Modes

Please refer to Figures 8, 9, and 12.

Proof state: 1. $((S \rightarrow Y) \vee (I \wedge Q))$, 2. $((I \wedge Q) \rightarrow D)$, 3. $\neg D$, 4. $((S \rightarrow Y) \rightarrow D)$, 5. $\neg(S \rightarrow Y)$, 6. $(I \wedge Q)$

Reasoning: Use *Modus Tollens* on $((S \rightarrow Y) \rightarrow D)$ and $\neg D$...

Reasoning error: Incorrect rule used

Solution context : D

Peer Feedback: "Great job on identifying *Modus Tollens* on $((S \rightarrow Y) \rightarrow D)$ and $\neg D$ as a key step. Now ask yourself, is there a step that could bridge your current progress to the **CONCLUSION** more efficiently..."

Figure 8: Example for **Vague Peer Feedback** - repeats student's incorrect solution

Proof state: 1. $(B \leftrightarrow \neg J)$, 2. $(\neg N \vee J)$, 3. $(B \vee \neg N)$, 4. $((B \rightarrow \neg J) \wedge (\neg J \rightarrow B))$, 5. $(N \rightarrow J)$, 6. $(\neg B \rightarrow \neg N)$

Student Reasoning: $(B \rightarrow \neg N)$ is optimal because it is the exact form of the conclusion, it can be derived from the contrapositive of $(\neg B \rightarrow \neg N)$...

Reasoning error: Incorrect derivation

Solution context: Derive $(J \vee \neg N)$ from $(\neg N \vee J)$ using *Commutative rule*.

Teacher Feedback: "Great job recognizing that $(B \rightarrow \neg N)$ is the contrapositive of $(\neg B \rightarrow \neg N)$ and using that to reach the conclusion..."

Figure 9: Example for **Teacher feedback validating incorrect reasoning**

Proof state:

$(S \rightarrow D) \vee I, (\neg S \vee Q) \rightarrow Y, \neg D, \neg D \rightarrow \neg I$

Student Reasoning: MT on $\neg D \rightarrow \neg I$ and $\neg D$ is needed to get $\neg I$ to be used later with $(S \rightarrow D) \vee I$ for DS

Solution Context: Derive $\neg S$ from $(S \rightarrow D)$ and $\neg D$ using *Modus Tollens* (MT).

Teacher Feedback: Great job identifying the relevant candidates! You correctly spotted that $(\neg D \rightarrow \neg I)$ and $\neg D$ could be useful, but there's a small mix-up with the rule. Recall: MT requires a negated *consequent*, while MP works with a *true antecedent*. How you might use MP on $(S \rightarrow D)$ and $\neg D$ to get $\neg D$.

Figure 10: Teacher (complete hint) feedback reveals solution on **incorrect** student response

Proof State (Excerpt)

$(S \rightarrow D), ((\neg S \vee Q) \rightarrow Y), \neg D, (\neg D \rightarrow \neg I), \dots$

Student Reasoning: $\neg S$ is optimal because we already have $(S \rightarrow D)$ from **INTERMEDIATE STEPS**, and simplifying it to $\neg S$ allows us to construct $(\neg S \vee Q)$...

Peer Solution context: $\neg S$

Peer Feedback: Excellent reasoning! You correctly identified how simplifying $(S \rightarrow D)$ helps set up the next step toward the conclusion...

Judge Feedback: Great job recognizing that you need $\neg S$ to proceed!, think about what rule you can use on $(S \rightarrow D)$ to turn it into a form more suitable for *Simp*...

Figure 11: Judge Verification Response, extending **Incorrect** Peer feedback (anchoring on **incorrect** student reasoning)

Proof state: 1. $\neg(\neg G \wedge B)$, 2. $(G \rightarrow D)$
Solution context: Derive $(\neg\neg G \vee \neg B)$ from $\neg(\neg G \wedge B)$ using DeM Law.
Student reasoning: Applying DN to $\neg(\neg G \wedge B)$ to get $(G \wedge B)$ is optimal because it directly gives us both G and B as available conjuncts...
Reasoning error: Incorrect rule application
Peer Feedback: "Great job applying double negation here! As you move forward, consider how you can use the new conjuncts to get closer..."
Judge Feedback: "You made a good attempt by considering double negation, but before applying DN, try using De Morgan's Law to rewrite..."

Figure 12: Example for **Judge Feedback repeating Peers' mistakes** - also directs towards the hint without recognizing the reasoning error

H.1 Student Agent Response

We show a representative Student agent response to demonstrate candidate generation, explicit reasoning, and next-step generation. The example corresponds to a single proof instance and highlights how intermediate reasoning is externalized for downstream tutoring and verification. The sample response is shown in Figure 13.

Givens: $((\neg K \vee L) \rightarrow (M \wedge N)), (K \rightarrow O), \neg O \neg K,$ $(\neg K \vee L), (M \wedge N)$
Conclusion: N
Student Reasoning: Applying a rule directly to $(M \wedge N)$ is the most efficient path to the conclusion. Since $(M \wedge N)$ is available, simplification yields the target.
Next Step: N
Rule: Simp
Parent Statement: $(M \wedge N)$

Figure 13: Sample problem instance and corresponding **correct** student response

I Manual Evaluation Protocol

To complement our automated metrics and better understand qualitative differences in feedback behavior across pipelines, we conducted a manual evaluation of a representative subset of model

responses. This analysis focuses on pedagogical properties of feedback that are difficult to capture through accuracy-based metrics alone, such as scaffolding quality and answer revelation.

I.0.1 Annotation Procedures

The samples were drawn from 200 distinct proof states stratified across solution categories and difficulty levels. Annotators included a graduate researcher and an instructor with expertise in the logic tutoring system, ensuring familiarity with the proof domain and pedagogical aspects of feedback. Annotators were provided with the problem context (givens, intermediates, and conclusion), the student's response, and the corresponding feedback generated by each feedback condition. Annotators worked independently and were blinded to model identity and quantitative performance results or feedback condition that generated the feedback. They were instructed to rate each feedback response independently on the four rubric dimensions based solely on the proof state, simulated solution, and generated feedback.

Annotators were trained on the rubric dimensions through an initial calibration phase using a shared sample of 20 pairs. Rubric definitions were iteratively refined, and disagreements were resolved through discussion until reaching a common interpretation of each dimension.

Due to the cognitive complexity of step-level logic feedback, during the initial calibration phase annotators agreed on a coarse 3-point ordinal scale (1–3) for all rubrics, with 1 = poor, 2 = partial, 3 = strong. This scale captures meaningful distinctions while maintaining annotation reliability and consistency. They then independently coded an additional sample until inter-rater reliability exceeded Cohen's $\kappa > 0.80$ per dimension and overall. Annotators then independently coded all remaining samples with no access to each other's labels or the research hypotheses.

I.1 Annotation Instructions

For each feedback response, rate independently on four dimensions: Correctness, Error Identification, Revealing, and Actionability. Each dimension is scored 1–3 as defined in the rubric. Base your rating solely on the proof state, the simulated student solution, and the generated feedback. For Correctness, assess whether the feedback contains logical errors that would mislead the student. For Error Identification, assess whether the feedback

precisely identifies the reasoning flaw or correctly affirms sound reasoning. For Revealing, assess the degree to which feedback discloses solution content explicitly. For Actionability, assess whether the feedback provides guidance specific enough for the student to proceed. When in doubt, refer to the calibration examples discussed during the calibration phase.

J Prompt Design

This section provides the complete prompts used for each agent in our multi-agent tutoring system. All prompts use structured JSON response formats to ensure consistent parsing.

Our prompt design translates established pedagogical principles into operational constraints for LLM agents. Each agent’s prompt encodes specific instructional strategies grounded in learning science research, ensuring that generated feedback aligns with effective tutoring practices. We describe the key design decisions below.

J.1 Student Prompt

The base system message includes the Role and Task information along with instructions, whereas the user message includes the problem instance itself. The prompt template is provided below in Figure 14.

J.2 Peer Prompt

The Peer operates under *minimal hint access*: it receives only the correct next step without knowledge of the complete solution path. The prompt enforces a critical **planning-before-feedback** requirement: the Peer must first generate an internal derivation plan explaining how the correct step is derived (identifying the rule and parent statements) before producing student-facing feedback. This design decision ensures the Peer develops a genuine understanding of the solution rather than pattern-matching, aligning with research showing that scaffolded feedback improves instructional quality (Serban et al., 2020). The prompt is illustrated in Figure 15.

J.3 Teacher Prompt

The Teacher agent has access to the complete next step and provides scaffolded feedback without revealing the answer. The prompt is illustrated in Figure 16.

J.4 Judge Prompt

The Judge agent is tasked with thoroughly evaluating both the student’s and the teacher’s responses to the logic proof problem. While the Teacher has access to the correct step and is therefore not asked to verify its own guidance, the Judge is specifically instructed to examine and identify any errors or inaccuracies present in both the student’s and the teacher’s responses.

Equipped with comprehensive access to the correct derivation steps from the knowledge base, the Judge carefully analyzes the reasoning behind both responses. Its role is to provide targeted evaluations and, where necessary, concise and actionable feedback to ensure alignment with the correct logic step. This approach ensures a robust layer of verification and strengthens the overall quality and pedagogical value of the feedback offered to the student. The prompt is illustrated in Figure 17.

Student Simulation Prompt

Role: You are a Student in an undergraduate Discrete Structures course solving a propositional logic proof. Your task is to produce the *single most optimal next step* that advances the proof toward the conclusion.

Task:

1. Review the givens and intermediate steps.
2. Propose 2–3 candidate next steps.
3. Select the candidate that most directly advances toward the conclusion.
4. Justify your choice and output the selected next step.

Constraints:

- Output exactly one next step in *symbolic notation only*.
- Use only predefined inference rules (e.g., MP, MT, Conj, DS).
- Parent statements must be actual expressions, not line numbers.

Response Format:

- CANDIDATES: 2–3 candidate steps with brief justification
- REASONING: Why the selected step is optimal
- NEXT_STEP: Symbolic expression
- RULE: Inference rule (short name)
- PARENT_STATEMENTS: Supporting expressions

Figure 14: Base System Prompt for Student

Peer Prompt

Role: You are a Peer evaluating a student's proposed next step in a propositional logic proof, with access to the KG-derived optimal step.

Task:

1. Analyze how the optimal step is derived (rule and parent statements).
2. Evaluate the student's candidates, reasoning, and chosen next step.
3. Classify the student's step as *Correct*, *Valid Alternative*, or *Incorrect*.
4. Provide brief, scaffolded feedback guiding the student toward the optimal step.

Constraints:

- Do not reveal the optimal step, its rule, or parent statements.
- Acknowledge what the student did correctly before addressing errors.
- Use Socratic questions to guide reasoning; keep feedback concise (2–3 sentences).
- Use predefined inference rule short names only.

Response Format:

- STUDENT_ERRORS: Brief explanation or Correct
- NEXT_STEP_CORRECTNESS: Correct / Suboptimal / Incorrect
- PEER_FEEDBACK: Scaffolded guidance without answer revelation

Figure 15: Base System prompt for Peer

Teacher Prompt

Role: You are a Teacher evaluating a student's proposed next step in a propositional logic proof, with access to the complete solution (KNOWLEDGE_BASE_STEPS).

Task:

1. Compare the student's response against the knowledge-base solution.
2. Identify errors in the student's logic, rule usage, or reasoning.
3. Classify the student's next step as *Correct*, *Valid Alternative*, or *Incorrect*.
4. Provide brief, scaffolded feedback guiding the student toward the correct solution.

Constraints:

- Do not reveal the exact next step, rule, or parent statements from the solution.
- Acknowledge correct aspects of the student's attempt before addressing errors.
- Use Socratic questions to guide reasoning; keep feedback concise (2–3 sentences).
- Refer to the student's candidates when relevant.
- Use predefined inference rule short names only.

Response Format:

- STUDENT_ERRORS: Brief explanation or Correct
- NEXT_STEP_CORRECTNESS: Correct / Suboptimal / Incorrect
- TEACHER_FEEDBACK: Scaffolded guidance without answer revelation

Figure 16: Base System prompt for Teacher

Judge (Verifier) Prompt

Role: You are an expert pedagogical AI Judge for propositional logic proof problems, with access to the complete solution (KNOWLEDGE_BASE_STEPS). You evaluate both the student's proposed next step and the Teacher's feedback.

Task:

1. Compare the student's response against the knowledge-base solution.
2. Identify errors in the student's reasoning, if any.
3. Classify the student's next step as *Correct*, *Valid Alternative*, or *Incorrect*.
4. Evaluate whether the Teacher's feedback correctly guides the student.
5. Either enhance the Teacher's feedback or override it with corrected guidance.

Constraints:

- Do not reveal the exact next step, rule, or parent statements from the solution.
- Acknowledge correct aspects of the student's attempt before addressing errors.
- Use Socratic questions to guide reasoning; scaffold rather than instruct.
- Override Teacher feedback if it is incorrect, misleading, or reveals the solution.
- Keep final feedback concise (2–3 sentences) and encouraging.
- Use predefined inference rule short names only.

Response Format:

- STUDENT_ERRORS: Brief explanation or Correct
- NEXT_STEP_CORRECTNESS: Correct / Suboptimal / Incorrect
- TEACHER_FEEDBACK_CORRECTNESS: Assessment of Teacher feedback
- JUDGE_ACTION: Enhanced or Overridden
- FINAL_FEEDBACK: Judge-approved scaffolded guidance

Figure 17: Base System prompt for Judge

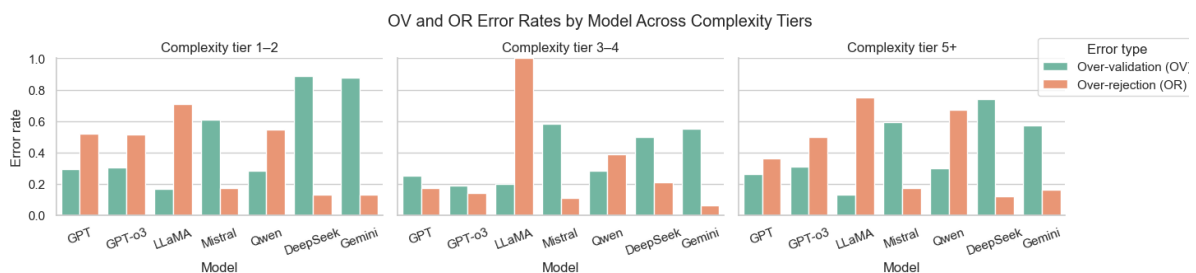


Figure 18: Over-validation (OV) and over-rejection (OR) rates by model across complexity tiers.

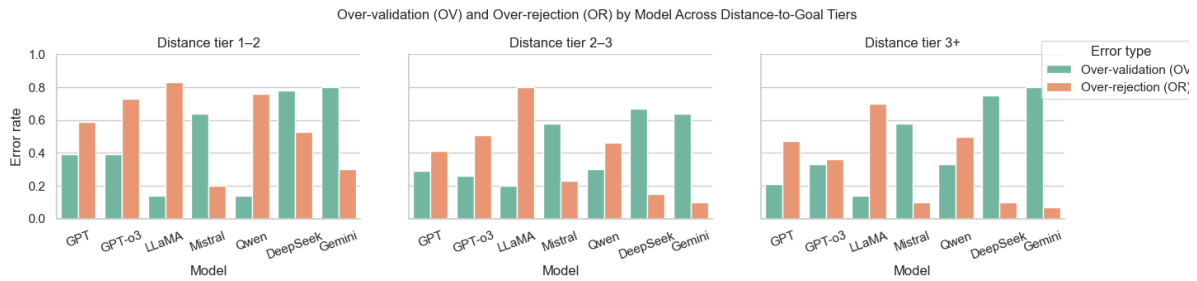


Figure 19: Over-validation (OV) and over-rejection (OR) rates by model across distance-to-goal tiers.

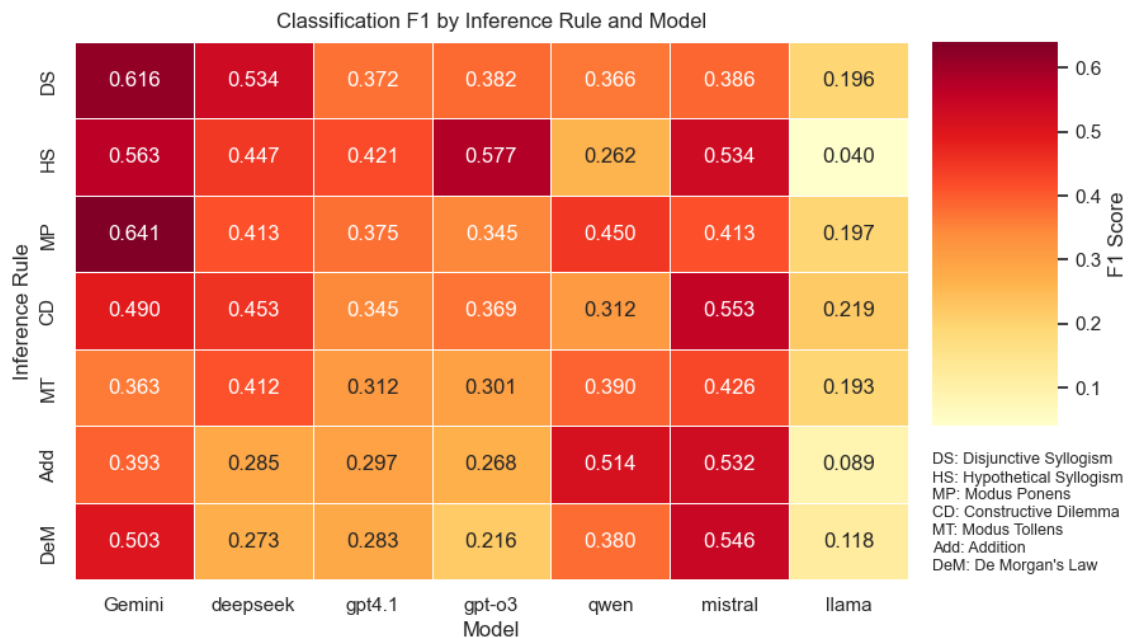


Figure 20: Classification F1 by inference rule and model. DS: Disjunctive Syllogism, HS: Hypothetical Syllogism, MP: Modus Ponens, CD: Constructive Dilemma, MT: Modus Tollens, Add: Addition, De Morgan's Law.

Retrieval-Augmented Tutoring for Algorithm Tracing and Problem-Solving in AI Education

Mragisha Jain¹, Tirth Bhatt¹, Griffin Pitts¹, Aum Pandya¹,
Peter Brusilovsky², Narges Norouzi³, Arto Hellas⁴, Juho Leinonen⁴, Bitu Akram^{1*}

¹North Carolina State University, ²University of Pittsburgh,

³University of California, Berkeley, ⁴Aalto University,

*Correspondence: bakram@ncsu.edu

Abstract

Students learning algorithms often need support as they interpret traces, debug reasoning errors, and apply procedures across unfamiliar problem instances. In this paper, we present KITE (Knowledge-Informed Tutoring Engine), a Retrieval-Augmented Generation (RAG)-based intelligent tutoring system designed to serve as a classroom teaching assistant for algorithmic reasoning and problem-solving tasks. KITE uses an intent-aware Socratic response strategy to tailor support to different student needs, responding with targeted hints, guiding questions, and progressive scaffolding intended to strengthen students' algorithmic problem-solving ability. To keep responses aligned with course content, KITE uses a multimodal RAG pipeline that retrieves relevant information from course materials. We evaluate KITE using three forms of assessment: RAGAs-based metrics for response grounding and quality, expert evaluation of pedagogical quality, and a simulated student pipeline in which a weaker language model interacts with KITE across two-turn dialogues and produces revised answers after receiving feedback. Results indicate that KITE produces contextually grounded and pedagogically appropriate responses. Further, using simulated students, KITE's feedback helped the student models produce more accurate follow-up responses on procedural and tracing questions, suggesting that its scaffolding can support algorithmic problem-solving. This work contributes a tutoring architecture and an evaluation approach for assessing retrieval-grounded explanations and scaffolded problem-solving feedback.

1 Introduction

Large language models (LLMs) such as ChatGPT are now widely used by students for learning support, including explanation, feedback, and problem-solving (Pitts et al., 2025b; Pitts and Motamedi, 2026). Students often value these tools because they provide immediate access to assistance when

instructors or teaching assistants are unavailable (Pitts et al., 2025b). Although these tools make information more accessible, prior work raises concerns that students may accept AI-generated responses without sufficient evaluation, especially when those responses appear complete and confident (Essel et al., 2024; Pitts et al., 2025c, 2026). In education, this can lead students to bypass the reasoning processes that assignments are designed to develop (Pitts et al., 2025c, 2026). These concerns highlight the need for LLM-based systems that provide timely, course-grounded information while delivering pedagogically appropriate support that helps students reason through learning tasks.

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) offers a promising approach for building course-grounded tutoring systems by allowing LLM responses to draw on curated instructional materials. This grounding can reduce unsupported or course-inconsistent claims and help align explanations with the concepts, terminology, and conventions used in a course. However, strong retrieval does not by itself ensure effective tutoring. Even when a system retrieves relevant material, it may still provide responses that are too direct, insufficiently instructional, or mismatched to the student's immediate learning need. Prior work on intelligent tutoring systems suggests that effective support depends on both the accuracy of the information provided and how assistance is delivered, including when to offer direct explanation, feedback, or more guided support (Koedinger and Alevan, 2007).

Socratic tutoring offers one way to address this challenge by guiding learners through targeted questions, prompts, and progressive hints instead of immediately providing full solutions. This approach is grounded in cognitive apprenticeship and guided facilitation (Collins et al., 1989; Hmelo-Silver and Barrows, 2006), and has been used in dialogue-based tutoring systems such as AutoTutor (Graesser et al., 1999). However, integrating

Socratic guidance into retrieval-grounded tutoring systems remains an open design problem: a course-specific tutor must stay faithful to instructional materials while also providing feedback that fits the type of problem the student is trying to solve.

In this paper, we present **KITE** (Knowledge-Informed Tutoring Engine), a RAG-based intelligent tutoring system that connects students to relevant course materials while using intent-aware tutoring strategies to support different forms of help-seeking. KITE uses a multi-stage multimodal retrieval pipeline to locate relevant instructional content and an intent-aware response strategy to determine how that content should be used in the response. For questions that require direct explanation, KITE provides responses aligned with retrieved course materials. For procedural, debugging, validation, and tracing questions, KITE provides targeted feedback, guiding questions, and progressive hints to support student reasoning. To evaluate KITE, we first assess its retrieval-grounded outputs for non-procedural questions using RAGAS-based metrics for grounding, relevance, and response quality. We then evaluate procedural and tracing questions through a simulated student pipeline in which a weaker language model revises its answers after receiving KITE’s feedback. Finally, human experts assess the resulting interactions to judge feedback quality and whether the revised answers show improvement. This work contributes (1) KITE, an intent-aware tutoring system that combines multimodal retrieval with pedagogical support, and (2) an evaluation of its retrieval-grounded responses and scaffolded feedback using automated metrics, simulated students, and expert evaluation. We assess two research questions: **RQ1**: How well does KITE produce grounded, course-relevant responses for non-procedural student questions? and **RQ2**: To what extent does KITE’s feedback support improved responses on procedural and tracing questions?

2 Related Work

RAG-based educational assistants have been used for a range of instructional purposes, including interactive learning support, content generation, and large-scale course deployment (Li et al., 2025). Across these systems, grounding LLM responses in course materials has generally improved factual accuracy compared to unaugmented models. For example, KAG (Hasan et al., 2025) reports Preci-

sion@5 of 0.85 and a 34% reduction in student task completion time, while MoodleBot (Neumann et al., 2024) achieves 88% accuracy on course-related queries. However, these systems primarily function as direct question-answering tools and do not adapt their responses to different forms of student help-seeking.

Although RAG can improve factual accuracy, deployment studies suggest that course-grounded assistants also need evaluation in instructional workflows. In one classroom deployment, students showed strong pre-exam engagement but declining adoption across cohorts, and 36.8% reported frustration when responses extended beyond a constrained knowledge base (Thesen and Park, 2025). Edison (Miroyan et al., 2025), a GPT-4-based RAG assistant deployed in a large data science course, showed that retrieving from course documents and historical Q&A can support factual and relevant responses to live student questions. The study also demonstrates the value of TA-in-the-loop evaluation, using instructor edits and ratings to assess factuality, relevance, style, and efficiency. EduMod-LLM (Mittal et al., 2026) extends this line of work by treating educational Q&A as a modular pipeline, separating function calling, retrieval, and response generation so that system behavior can be evaluated more transparently.

Dialogue-based tutoring provides another foundation for supporting student reasoning. AutoTutor (Graesser et al., 1999) showed that progressive hints and collaborative answer refinement produced dialogues rated above the “good” threshold by domain experts, with semantic evaluation correlating at 0.49 with expert judgment. More recently, (Li et al., 2026) reported significant gains in self-efficacy ($d = 0.57$) from a Socratic AI platform in healthcare education. LeanTutor (Patel et al., 2026) similarly emphasizes guided feedback by combining LLMs with a theorem prover to check student proofs, identify errors, and provide hints toward a correct proof without giving away the complete answer. These systems show the value of scaffolded feedback for learning tasks that require students to reason through a process, yet they do not incorporate retrieval grounding to keep responses aligned with course-specific materials.

Other systems explore how retrieval and response strategies can be adapted for learning contexts. LPITutor (Liu et al., 2025) supports adaptive difficulty modulation through RAG and prompt engineering. KG-RAG (Dong et al., 2025) combines

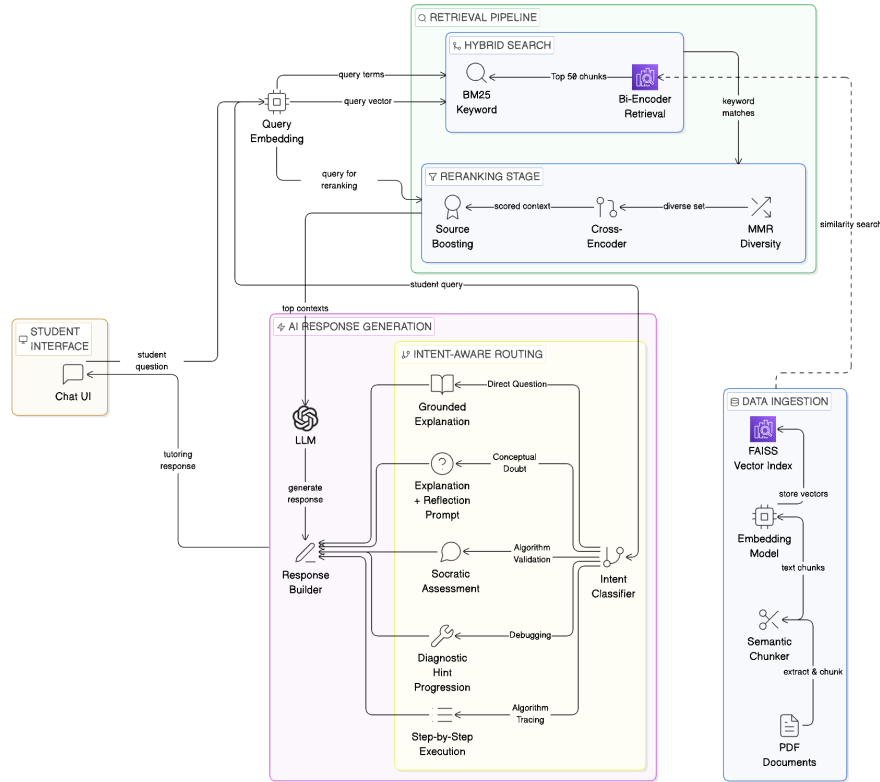


Figure 1: KITE architecture

semantic retrieval with an expert-validated knowledge graph and reports a 35% improvement in learning outcomes ($d = 0.86$) in a study of 76 students, though its reliance on manual expert validation limits scalability. AutoTA (Dahal et al., 2025) provides a related approach to intent-aware educational assistance by classifying student queries and routing them to specialized response strategies. These systems show the value of adapting retrieval, domain structure, and response behavior to different learning needs. KITE builds on this direction by pairing multi-stage retrieval with intent-aware tutoring strategies for algorithmic reasoning tasks.

3 System Design

KITE is a retrieval-augmented tutoring system designed to support course-grounded dialogue for algorithmic reasoning and problem-solving tasks. As shown in Figure 1, the system includes five components: document preprocessing, embedding generation, multi-stage retrieval, intent-aware response generation, and session management.

3.1 Phase 1: Document Ingestion and Preprocessing

KITE begins by extracting text from course PDFs using *PyMuPDF*. Extraction is performed page by

page so that the original document structure remains traceable during retrieval. To reduce noise before indexing, the system applies a **frequency-based** cleaning procedure that removes repeated headers, footers, page numbers, and other formatting artifacts. Specifically, it examines the first and last two lines of each page, identifies repeated patterns that occur across pages, removes those patterns along with page numbers and special characters, and normalizes whitespace.

The cleaned text is then segmented into semantically coherent chunks for retrieval. We use **section-aware chunking** with a target size of 500 characters, or about 125 tokens, and a 100-character overlap. Headers are identified and retained to preserve local structure, while overlap carries forward the final two sentences of the preceding chunk.

3.2 Phase 2: Embedding Generation

Each chunk is encoded using OpenAI’s text-embedding-3-large model, producing 3072-dimensional embeddings. These vectors are L2-normalized so cosine similarity reflects semantic direction and are stored in a FAISS index (Johnson et al., 2019) for efficient local retrieval.

3.3 Phase 3: Multi-Stage Retrieval Pipeline

KITE uses a multi-stage pipeline designed to balance high recall and precision in retrieving course content. Retrieval begins with a dense bi-encoder search that returns the top 50 candidate chunks for a given student query. The query and document chunks are encoded independently, and similarity is computed using cosine similarity, allowing the system to capture semantically related content.

The candidate set is then refined through hybrid retrieval. Dense similarity contributes 70% of the retrieval score, while sparse BM25 keyword matching contributes 30%. This combination captures both semantic similarity and exact lexical overlap, which is useful when students use course-specific terminology, notation, or algorithm names.

To reduce redundancy among retrieved passages, KITE applies Maximal Marginal Relevance (MMR) with λ set to 0.7:

$$MMR = \lambda \times \text{Relevance} + (1 - \lambda) \times \text{Diversity}$$

The retrieved candidates are reranked using a cross-encoder/ms-marco-MiniLM-L-6-v2 reranking model implemented through Sentence Transformers, where the query and document are jointly encoded to produce more precise relevance scores. Finally, KITE applies source-based boosting so that chunks from official course materials receive higher priority. Chunks with reranking scores above 0.6 receive an additional boost of 0.3. The final context passed to the generator consists of the top eight retrieved chunks.

3.4 Phase 4: Intent Classification and Pedagogical Response Generation

KITE does not use a single response strategy for all student questions. Instead, it first classifies each query by pedagogical intent and then generates a response that matches the instructional purpose of the interaction. This allows the system to distinguish among questions, debugging requests, and other forms of help-seeking.

3.4.1 Intent Classification

Each incoming query is classified into one of five pedagogical intents using a keyword and pattern-matching classifier, as shown in Figure 1.

- **Direct Question:** factual queries seeking definitions or explanations (e.g., “What is A*?”).

- **Conceptual Questions:** deeper *why* or *how* questions probing understanding (e.g., “Why does BFS guarantee shortest path?”).
- **Algorithm Validation:** queries where a student submits their own implementation or trace for assessment.
- **Debugging:** queries involving a specific error or incorrect output.
- **Algorithm Tracing:** requests to step through the execution of an algorithm on a concrete problem instance (e.g., “Trace A* on this graph starting from node S”).

The classified intent determines which response generation strategy is invoked. In addition to these five query intents, KITE includes a dedicated answer evaluation mode for cases in which a student submits a written answer for assessment. This mode bypasses intent classification and routes directly to the feedback generation pipeline.

3.4.2 LLM Generation and Intent-Aware Response Strategy

All response generation in KITE is handled by GPT-5. Outputs are grounded in a structured prompt that injects the top eight retrieved chunks into a [CONTEXT] block. The model is instructed to prioritize course materials and avoid introducing information that is unsupported by the retrieved context, helping keep responses aligned with the course.

For direct questions and conceptual doubts, KITE produces explanations grounded in the retrieved material. Responses are written in a tutor-like tone that emphasizes reasoning instead of brief answer delivery. For conceptual doubts, the response also includes a follow-up question intended to prompt reflection.

For algorithm validation tasks, KITE adopts a Socratic assessment strategy instead of directly identifying errors. Responses include a brief evaluation of the student’s approach, acknowledgement of correct components, and guiding questions that target specific issues. This design supports learning without explicitly revealing the final solution, encouraging students to continue working through the problem independently.

For debugging assistance, KITE generates diagnostic prompts that guide students toward identifying errors through self-examination. Each response follows a structured hint progression and includes a learning point that connects the observed bug to the underlying conceptual principle, reinforcing understanding beyond the immediate correction.

For algorithm tracing queries, KITE retrieves the relevant procedural steps and rules from course materials and applies them step by step to the student’s specific problem instance. Each step explicitly maintains and updates algorithmic state variables such as OPEN lists, CLOSED sets, and selected nodes, following the tie-breaking rules and constraints defined in the query. The response concludes with the final path and cost when applicable.

3.5 Phase 5: Session Management

KITE maintains session state across multi-turn interactions to preserve continuity within a conversation. For each session, the system stores the original query, detected intent, prior responses, and any hints provided. When a student submits a follow-up query, KITE uses this stored context to determine how the interaction should continue: related direct and conceptual questions are treated as follow-ups, while validation, debugging, and tracing requests remain within their intent-specific response strategy when they concern the same problem or algorithm. For these turns, KITE constructs a brief context summary from the prior interaction and appends it to the retrieval prompt to reduce repetition and support progressive guidance.

4 Methodology

To evaluate KITE, we use three forms of assessment. First, we examine non-procedural responses using RAGAs-based metrics for grounding, relevance, and answer quality against instructor-authored reference answers. We then use a simulated student pipeline to assess whether KITE’s feedback helps produce improved responses on procedural and tracing questions. Finally, experts evaluate the pedagogical quality of KITE’s feedback and the resulting answer revisions.

4.1 Evaluation Dataset

We constructed an evaluation dataset of 109 questions drawn from the lecture slides and textbook used in a university *Introduction to AI* course, with each question paired with an instructor-verified reference answer. The dataset included 42 algorithmic questions, 51 procedural questions, and 16 direct-retrieval questions. We applied RAGAs to the 58 non-procedural questions, consisting of the algorithmic and direct-retrieval subsets, because these responses can be evaluated against reference answers for grounding, relevance, and answer quality.

Questions requiring procedural reasoning or algorithm tracing were evaluated separately through the simulated student pipeline and expert review described in Section 4.3.

4.2 RAGAs Evaluation

We evaluate KITE’s non-procedural responses using the RAGAs framework (Es et al., 2024; Roychowdhury et al., 2024), reporting six metrics:

- **Faithfulness:** Measures whether statements in the generated response are supported by the retrieved context, computed as the proportion of answer claims judged to be grounded in the retrieved chunks.
- **Answer Relevance:** Measures how well the generated response addresses the original question, computed from the cosine similarity between the user’s question and questions generated from the response.
- **Context Relevance:** Measures how much of the retrieved context is relevant to answering the question, computed as the proportion of retrieved statements judged to be useful.
- **Answer Similarity:** Measures semantic similarity between the generated response and the instructor-authored reference answer using sentence embeddings.
- **Factual Correctness:** Measures factual agreement between the generated response and the reference answer using an F1 score over claims classified as true positives, false positives, and false negatives.
- **Answer Correctness:** Measures overall correctness of the generated response relative to the reference answer as a weighted combination of factual correctness (0.75) and answer similarity (0.25).

All metrics use gpt-4o-mini as the judge model and text-embedding-3-small for similarity metrics. Retrieval uses top_k=5 from an initial candidate pool of 50.

4.3 Simulated Student Evaluation and Expert Evaluation

For procedural and algorithm-tracing questions, KITE provides Socratic feedback and guidance, making standard automatic scoring less appropriate. To evaluate how well this feedback supports learning-oriented revision, we use a two-stage simulated student pipeline followed by expert review.

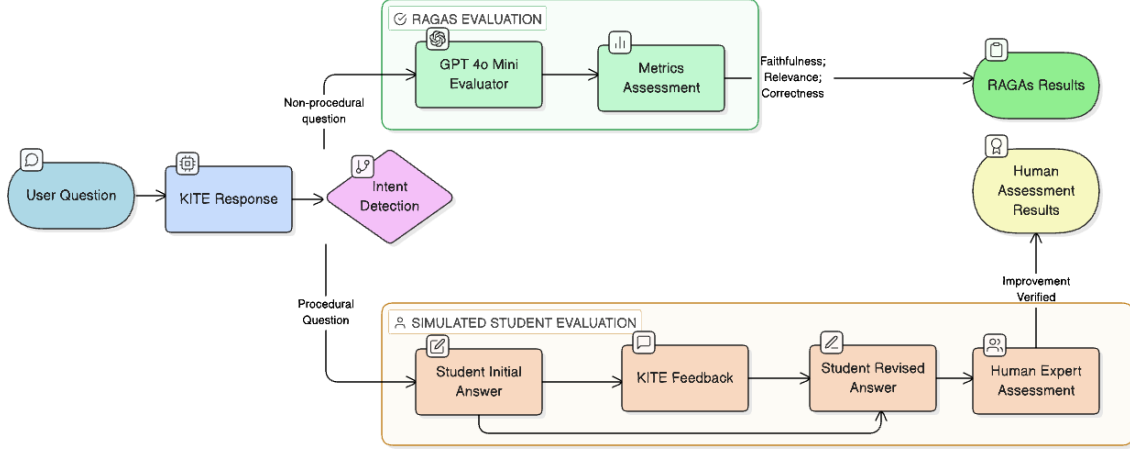


Figure 2: Evaluation pipeline

Simulated Student Pipeline. Building on prior work that uses simulated student-tutor interactions to evaluate pedagogical support (Dinucu-Jianu et al., 2025), we use Meta-Llama-3.1-70B-Instruct as a proxy student in a structured interaction with KITE:

1. **Round 1:** The student model answers each question without assistance, establishing an unaided baseline.
2. **KITE Feedback:** KITE evaluates the student’s answer and provides feedback intended to guide revision.
3. **Round 2:** The student model revises its answer using KITE’s feedback.

Expert Evaluation. Three experts reviewed each interaction set, consisting of the Round 1 answer, KITE’s feedback, and the Round 2 answer. They judged whether the revised response demonstrated improved correctness and reasoning, and evaluated the quality of KITE’s feedback using a structured rubric adapted from prior work (Pauzi et al., 2025).

The rubric includes three dimensions. *Mistake Remediation* assesses whether the tutor correctly identifies the student’s error and explicitly acknowledges it in the response. *Scaffolding and Guidance* assesses whether the tutor provides appropriate support without revealing the answer and offers clear next-step direction. *Coherence and Tone* assesses whether the dialogue reads naturally and maintains an encouraging and supportive tone. Each criterion is scored as Yes/No, with NA used when a criterion is not applicable.

5 Results

5.1 RAGAs Evaluation

Table 1 reports the six RAGAs metrics evaluated on the 58 non-procedural questions, consisting of 42 algorithmic questions and 16 direct-retrieval questions.

Metric	Mean	Std. Dev.
Faithfulness	0.8486	0.2103
Answer Relevance	0.7558	0.2032
Context Relevance	0.9352	0.1905
Answer Similarity	0.7586	0.0923
Factual Correctness	0.4483	0.2477
Answer Correctness	0.6363	0.1810

Table 1: RAGAs evaluation summary ($n = 58$).

KITE performs strongly on retrieval and grounding metrics. Faithfulness (0.85) indicates that most answer statements are supported by the retrieved context, while context relevance (0.94) shows that the retrieved passages are highly pertinent to the question. Answer relevance (0.76) and answer similarity (0.76) further suggest that KITE’s responses remain on-topic and semantically aligned with instructor-authored reference answers.

Factual correctness (0.45) is lower than the other RAGAs measures. As discussed in Section 7, this metric is sensitive to claim-level overlap with a single reference answer and may understate the quality of responses that are accurate but phrased differently or provide additional valid detail. For this reason, answer similarity is used as the primary indicator of response quality in this setting. Its low

variance ($\sigma = 0.09$) also suggests relatively consistent performance across the evaluated questions.

5.2 Simulated Student and Expert Evaluation

Table 2 summarizes the expert rubric scores for 44 simulated student–KITE interaction triples. Inter-rater agreement between the two expert annotators was high, with Cohen’s $\kappa = 0.88$ and a raw agreement rate of 98.15%, indicating strong consistency in rubric judgments.

Metric	% Yes	% No	% N/A
Mistake Remediation (Identifying)	63.63	6.82	29.55
Mistake Remediation (Acknowledging)	63.63	6.82	29.55
Scaffolding	93.18	6.82	—
Guidance	93.18	6.82	—
Coherence (Naturalness)	93.18	6.82	—
Tone (Encouraging)	93.18	6.82	—

Table 2: Expert evaluation rubric results ($n = 44$).

KITE receives consistently high ratings for scaffolding, guidance, coherence, and tone, with 93.18% Yes judgments on each dimension. These results indicate that its feedback is generally well-structured, actionable, and supportive throughout the interaction. Mistake remediation receives 63.63% Yes judgments, but 29.55% of cases are marked N/A because the simulated student’s initial response was already correct and no error identification was required. When remediation is applicable, the results indicate that KITE identifies and acknowledges student errors appropriately.

Answer Improvement. Table 3 reports how students responses changed from Round 1 to Round 2 after receiving KITE’s feedback. The transition labels reflect expert judgments of whether responses were Incorrect, Partially Correct, or Correct with respect to the course materials. Among the 27 interactions that were not already correct, 24 showed improvement after KITE’s feedback (88.89%).

The most common transition was from Partially Correct to Correct, occurring in 14 cases (31.82%). This suggests that KITE is particularly effective at helping students resolve remaining reasoning gaps in responses that are already moving in the right direction. In six additional cases (13.63%), the response remained Partially Correct but still improved in quality, indicating that KITE’s feedback can support meaningful revision even when the student model does not reach a correct answer.

Transition	Count	%
Incorrect → Correct	1	2.27
Incorrect → Partially Correct	3	6.82
Already Correct	17	38.64
Partially Correct → Correct	14	31.82
Partially Correct → Partially Correct with Improvement	6	13.63
N/A	3	6.82

Table 3: Answer improvement breakdown ($n = 44$).

6 Discussion

This study examined whether a course-grounded, intent-aware tutoring system could provide reliable retrieval-based support and pedagogically useful feedback for problem-solving tasks. The results are encouraging with regard to both aims. For RQ1, KITE’s faithfulness (0.85) and context relevance (0.94) indicate that its responses are closely grounded in retrieved course material, while answer similarity (0.76) shows consistent alignment with instructor-authored reference answers. For RQ2, among the 27 simulated-student interactions with KITE, in which the initial student response was not already correct, experts judged 24 revised answers (88.89%) as improved after receiving KITE’s feedback. This suggests that KITE’s feedback provided guidance the student model could use to correct or strengthen its reasoning in a follow-up response. Experts rated 93.18% of KITE’s feedback positively for scaffolding and guidance, further indicating that the feedback was instructionally purposeful and well-structured.

Retrieval and response quality. The RAGAs evaluation indicated that KITE performed well on measures tied to retrieval and grounding, with context relevance of 0.94 and faithfulness of 0.85. These results show that KITE’s multi-stage retrieval pipeline surfaced course-specific material relevant to the questions and that its responses remained closely grounded in that retrieved context. At the same time, the lower factual correctness score (0.45) warrants careful interpretation, particularly relative to answer similarity (0.76). Prior work has noted that RAGAs-style claim matching and score stability can vary across response formulations and evaluation settings (Roychowdhury et al., 2024; Antal and Buza, 2025). In light of prior work, the 0.31-point gap observed in our results may reflect limitations of factual correctness as a reference-based metric for evaluating KITE’s pedagogically framed responses, although our evaluation does not isolate the source of that discrepancy.

Pedagogical effectiveness and design implications. Following the RAGAs evaluation, the simulated student and expert evaluations examined whether KITE’s feedback supported stronger revised answers, and was judged to be pedagogically appropriate. Specifically, the 88.89% improvement rate in the simulated student pipeline, together with the strong expert rubric scores for scaffolding, guidance, coherence, and tone, support that KITE’s feedback can support stronger follow-up answers on procedural and tracing questions. This emphasis on feedback quality is consistent with prior survey work on LLM applications in programming education, which argues that the educational value of these systems depends on aligning model capabilities with pedagogical goals, including the use of scaffolding and feedback strategies (Pitts et al., 2025a). In KITE, this alignment is reflected in pairing retrieval-grounded generation with feedback strategies designed for different forms of student support, such as direct explanations, or algorithmic-tracing guidance. While these findings are encouraging, the current evaluation design limits their generalizability, as discussed in Section 6.1.

6.1 Limitations and Future Work

This study has limitations that motivate future work. First, our evaluation of factual correctness is constrained by limitations of the RAGAs framework. KITE produces pedagogically framed explanations that may paraphrase or elaborate on course material, whereas RAGAs decomposes each response into atomic claims and uses NLI-style entailment to assess agreement with a single instructor-authored answer. With this, responses that are semantically aligned with the expected answer but differ in phrasing, detail, or framing may receive lower factual correctness scores. The 0.31-point gap between factual correctness (0.45) and answer similarity (0.76) in Table 1 is consistent with this concern, although our evaluation does not isolate the source of that discrepancy. Answer similarity, which is based on semantic embeddings, remains substantially higher and shows low variance ($\sigma = 0.09$), indicating relatively stable semantic alignment across questions. We therefore treat answer similarity as the primary quality indicator and note that using multiple human-written answers could reduce this limitation in future evaluations. Related concerns have been noted in prior work: Roychowdhury et al. (Roychowdhury et al., 2024) discuss limita-

tions in how RAGAs decomposes and assigns statements during metric computation, while Antal and Buza (Antal and Buza, 2025) show that RAGAs-based evaluation outcomes vary across question types and retrieval conditions. Future evaluations could use richer answer sets and metrics.

Second, the simulated student pipeline relies on a single LLM, Meta-Llama-3.1-70B-Instruct, as a proxy for student behavior. As a result, improvement between Round 1 and Round 2 should be interpreted as evidence that KITE’s feedback makes a stronger answer more recoverable, not as direct evidence of genuine student learning. Real learners may differ substantially in both the magnitude and pattern of improvement. Although this design is useful for early-stage evaluation, it cannot substitute for classroom evidence. A necessary next step is deployment with real students, including pre- and post-interaction assessments, analysis of revision behavior over time, and closer examination of how learners engage with feedback.

Third, the expert evaluation covers a relatively limited set of interaction cases, which constrains the generalizability of the findings. Although inter-rater agreement was strong ($\kappa = 0.88$), judgments of answer improvement and pedagogical quality still involve subjectivity, and the sample size limits precision. Future work should expand the annotation set and use finer-grained scoring schemes to better capture variation in feedback quality and answer improvement across question and error types.

7 Conclusion

We presented KITE, a RAG-based intelligent tutoring system that combines a five-stage retrieval pipeline with intent-aware pedagogical response generation. KITE adapts its responses to the type of student query, providing grounded explanations for factual questions and Socratic scaffolding for procedural and reasoning tasks. To evaluate these response modes, we introduced a two-part evaluation framework. RAGAs metrics assess retrieval quality, while a simulated student pipeline examines whether KITE’s feedback supports improved responses on procedural and tracing questions. Expert review using a structured rubric further evaluates the pedagogical quality of KITE’s feedback and verifies improvement in students’ revised answers. This work contributes an intent-aware tutoring architecture and an evaluation approach for RAG systems with mixed response strategies.

Acknowledgements

This research was supported by the U.S. National Science Foundation (NSF) under Grant #2426837. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect views of the NSF. This work was additionally supported by Research Council of Finland grants #356114 and #367787.

References

- Margit Antal and Krisztian Buza. 2025. Evaluating open-source llms in rag systems: a benchmark on diploma theses abstracts using ragas: M. antal, k. buza. *Acta Universitatis Sapientiae, Informatica*, 17(1):5.
- Allan Collins, John Seely Brown, and Susan E. Newman. 1989. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In Lauren B. Resnick, editor, *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, pages 453–494. Lawrence Erlbaum Associates.
- Rajashree Dahal, Greg Murray, Robin Chataut, Mohamed Hefaida, Anurag Srivastava, and Prashna Gyawali. 2025. Autota: A dynamic intent-based virtual teaching assistant for students using open source llms. *IEEE Access*.
- David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. From problem-solving to teaching problem-solving: Aligning llms with pedagogy using reinforcement learning.
- Chenxi Dong, Yimin Yuan, Kan Chen, Shupeu Cheng, and Chujie Wen. 2025. How to build an adaptive ai tutor for any course using knowledge graph-enhanced retrieval-augmented generation (kg-rag). In *2025 14th International Conference on Educational and Information Technology (ICEIT)*, pages 152–157. IEEE.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th conference of the european chapter of the association for computational linguistics: system demonstrations*, pages 150–158.
- Harry Barton Essel, Dimitrios Vlachopoulos, Albert Benjamin Essuman, and John Opuni Amankwa. 2024. Chatgpt effects on cognitive skills of undergraduate students: Receiving instant responses from ai-based conversational large language models (llms). *Computers and Education: Artificial Intelligence*, 6:100198.
- Arthur C Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Roger Kreuz, Tutoring Research Group, and 1 others. 1999. Autotutor: A simulation of a human tutor. *Cognitive Systems Research*, 1(1):35–51.
- Hadi Hasan, Ali Ismail, Ammar Mohanna, and Ali Chehab. 2025. Kag: A scalable knowledge-augmented generation system for educational content management. In *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*, pages 503–508. IEEE.
- Cindy E Hmelo-Silver and Howard S Barrows. 2006. Goals and strategies of a problem-based learning facilitator. *Interdisciplinary journal of problem-based learning*, 1(1):4.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7(3):535–547.
- Kenneth R Koedinger and Vincent Aleven. 2007. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational psychology review*, 19(3):239–264.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Yan Li, Janelle Yorke, Jiaying Li, Mengting He, Yushen Dai, Yan Zhao, Jing Qin, and Xiangen Hu. 2026. An innovative socratic method-based artificial intelligence platform for healthcare education: A quasi-experimental study. *Nurse education in practice*, page 104770.
- Zongxi Li, Zijian Wang, Weiming Wang, Kevin Hung, Haoran Xie, and Fu Lee Wang. 2025. Retrieval-augmented generation for educational application: A systematic survey. *Computers and Education: Artificial Intelligence*, 8:100417.
- Zhensheng Liu, Prateek Agrawal, Saurabh Singhal, Vishu Madaan, Mohit Kumar, and Pawan Kumar Verma. 2025. Lpitutor: an llm based personalized intelligent tutoring system using rag and prompt engineering. *PeerJ Computer Science*, 11:e2991.
- Mihran Miroyan, Chancharik Mitra, Rishi Jain, Gireeja Ranade, and Narges Norouzi. 2025. Analyzing pedagogical quality and efficiency of llm responses with ta feedback to live student questions. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 770–776.
- Meenakshi Mittal, Rishi Khare, Mihran Miroyan, Chancharik Mitra, and Narges Norouzi. 2026. Edumod-llm: A modular approach for designing flexible and transparent educational assistants. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 40652–40660.

- Alexander Tobias Neumann, Yue Yin, Sulayman Sowe, Stefan Decker, and Matthias Jarke. 2024. An llm-driven chatbot in higher education for databases and information systems. *IEEE Transactions on Education*, 68(1):103–116.
- Manooshree Patel, Rayna Bhattacharyya, Thomas Lu, Arnav Mehta, Niels Voss, Narges Norouzi, and Gireeja Ranade. 2026. Leantutor: Towards a verified ai mathematical proof tutor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 40670–40678.
- Zaki Pauzi, Michael Dodman, and Manolis Mavrikis. 2025. Automating pedagogical evaluation of llm-based conversational agents. In *Ceur Workshop Proceedings*, volume 4006. CEUR.
- Griffin Pitts, Anurata Prabha Hridi, and Arun Balajee Lekshmi Narayanan. 2025a. A survey of llm-based applications in programming education: Balancing automation and human oversight. In *Proceedings of the Fourth Workshop on Bridging Human-Computer Interaction and Natural Language Processing (HCI+ NLP)*, pages 255–262.
- Griffin Pitts, Viktoria Medvedeva Marcus, and Sanaz Motamedi. 2025b. Student perspectives on the benefits and risks of ai in education. In *2025 ASEE Annual Conference & Exposition*.
- Griffin Pitts and Sanaz Motamedi. 2026. What drives students’ use of ai chatbots? technology acceptance in conversational ai. *arXiv preprint arXiv:2602.20547*.
- Griffin Pitts, Neha Rani, and Weedguet Mildort. 2026. Trust and reliance on ai in education: Ai literacy and need for cognition as moderators. *arXiv preprint arXiv:2604.01114*.
- Griffin Pitts, Neha Rani, Weedguet Mildort, and Eva-Marie Cook. 2025c. Students’ reliance on ai in higher education: identifying contributing factors. In *International Conference on Human-Computer Interaction*, pages 86–97. Springer.
- Sujoy Roychowdhury, Sumit Soman, HG Ranjani, Neeraj Gunda, Vansh Chhabra, and SAI KRISHNA BALA. 2024. Evaluation of rag metrics for question answering in the telecom domain. In *ICML 2024 Workshop on Foundation Models in the Wild*.
- Thomas Thesen and Soo Hwan Park. 2025. A generative ai teaching assistant for personalized learning in medical education. *NPJ Digital Medicine*, 8(1):627.

LLM-Powered but Rule-Grounded: Pedagogically Relevant Grammatical Error Characterization for Learner Model Construction

Soroosh Akef^{1,2,3} Amália Mendes² Patrick Rebuschat^{3,4} Detmar Meurers^{1,3}

¹Leibniz Institute für Wissensmedien (IWM), Germany

²Center of Linguistics, Faculty of Arts and Humanities, University of Lisbon, Portugal

³LEAD Graduate School and Research Network, University of Tübingen, Germany

⁴Lancaster University, United Kingdom

s.akef@iwm-tuebingen.de d.meurers@iwm-tuebingen.de

mendes@edu.ulisboa.pt p.rebuschat@lancaster.ac.uk

Abstract

Grammatical error correction approaches rarely characterize the pedagogical value of corrected errors. We propose a framework that combines LLM-based second-language writing correction with a rule-based characterization module to identify pedagogically relevant, fine-grained grammatical properties in learner texts. The characterization module targets 252 European Portuguese properties which are categorized by the CEFR level at which they are taught according to an authoritative curriculum, and property accuracy is inferred from contrasts between the learner and corrected texts. We evaluate the framework extrinsically by training interpretable automatic proficiency assessment models on accuracy features extracted from characterized errors in a Portuguese learner corpus. Across different prompting strategies, we show that models trained on features derived from LLM-corrected texts perform similarly to those trained on features derived from annotator-corrected texts and comparably to models trained on linguistic complexity features. Feature importance overlap is likewise high, and similar predictive patterns are observed in both annotator-based and LLM-based models, further supporting the validity of the proposed framework.

1 Introduction

Grammatical error correction (GEC) has received significant attention from the ACL community in recent years, as evidenced by the organization of multiple shared tasks (Ng et al., 2013; Bryant et al., 2019; Masciolini et al., 2025). The attention this task has received is well justified, as a system capable of proofreading free prose is not only of interest to second language (L2) learners but also to native (L1) speakers.

Identifying errors in learners' production is particularly relevant in the context of L2 learning, as it can be used not only to give feedback to learners,

but also to construct learner models that can be used in intelligent tutoring systems (ITSs) and intelligent computer-assisted language learning (ICALL) systems to deliver personalized language education. However, while GEC has moved toward the direction of evaluation methods with greater emphasis on characterizing errors at a finer level of granularity, the current granularity level of error annotation is insufficient for a learner model built based on the identification of such errors to be actionably informative. For instance, while the ERRANT scorer (Bryant et al., 2017) used to evaluate models submitted to the BEA-2019 shared task (Bryant et al., 2019) distinguishes between 25 main error categories, the information that an error represents the VERB:TENSE category is not fine-grained enough to build and update a learner model that can personalize L2 learning.

Evaluation methods used in GEC, and how validly they assess whether a corrected text contains corrections that are informative about an L2 learner's language development, also need to be reconsidered in the context of learner modeling. While reference-based evaluation methods compare the corrected text with a reference corrected by an expert, they do not account for the reality that there could be multiple valid approaches to correct a text (Bryant et al., 2023). To mitigate the limited nature of reference-based evaluation methods, various reference-free evaluation methods have been proposed (Asano et al., 2017; Choshen and Abend, 2018; Yoshimura et al., 2020; Islam and Magnani, 2021; Maeda et al., 2022; Östling et al., 2025). However, the existing reference-free evaluation methods also do not approximate the pedagogical relevance of the corrections in L2 learning contexts. Relatedly, Masciolini et al. (2025) distinguish between minimal edits and fluency edits in the shared task MultiGEC-2025, the latter of which emphasizes rephrasing the original text in favor of idiomaticity. The preference of one GEC approach

over the other in an L2 context likely hinges on the aspects of the L2 which are to be monitored and acted upon in a learner model.

The present work thus seeks to address the twofold challenge of identifying pedagogically relevant grammatical errors from texts produced by L2 European Portuguese learners and establishing the extent to which the identified errors are informative about L2 language development and can therefore be validly used to construct and update a learner model representing the learner's knowledge of various grammatical properties.

2 Related Work

A learner model represents a learner's current knowledge and mastery of a particular domain, inferred based on the data collected from the learner's interaction with a learning system (Xu and Bull, 2010). Learner models are thus vital to the construction of any adaptive educational system, as only through a detailed and accurate understanding of learners' competencies can they be matched to developmentally proximal learning material.

ICALL systems capable of identifying learners' errors and constructing or updating learner models accordingly have traditionally depended on mal-rules, symbolic representations of various ill-formed productions by learners according to an error taxonomy (Schneider and McCoy, 1998).

To build the BELLOC system, intended for English-speaking learners of French, Chanier et al. (1992) took advantage of what they referred to as "applicable rules," which explored rules underlying a learner's misconceptions if a correct parse for the sentence produced by the learner was not found. A limitation of the system was that when multiple potential misconceptions were matched, the system would prompt the learner to assess the grammaticality of test cases intended to isolate the learner's misconception, consequently exposing learners to further ill-formed constructions, which could reinforce the misconceptions.

A similar learner modeling strategy was adopted for the ICALL system Mr. Collins (Bull, 1994), which utilized pronoun placement tasks targeting European Portuguese clitic usage that required students to position pronouns and apply necessary phonetic contractions. The system addressed the issue of isolating a learner's misconceptions by taking into account the acquisition order of the structures. When a grammatical parse of a sentence

failed, the system first attempted to identify target-language misconceptions, such as overgeneralization. If that failed, the system investigated negative transfer, prioritizing background languages based on the student's proficiency and linguistic distance to Portuguese. The limited scope of the grammatical structures targeted, as well as the controlled nature of the tasks, is among the limitations of this system.

Another relevant system was German Tutor (Heift and McFetridge, 1999), later renamed to E-Tutor (Heift, 2005, 2008), which incorporated a learner model that tracked a student's performance across various grammatical properties. Each property was assigned a score from 0 to 30, where 0 indicated an expert and 30 a novice. Unlike previous systems that relied on mal-rules after a parse failure, this system utilized a grammar that freely generated parses for both grammatical and ungrammatical input by relaxing specific constraints. With every successful production, the score representing the property in the learner model was decremented, and with every failed production, it was incremented. Finally, an error priority queue ensured that the learner received feedback on only the most significant error depending on their performance history or the instructor's decision to avoid pedagogical overload.

The ICICLE system (Michaud et al., 2005) further extended these approaches by modeling the evolution of interlanguage in deaf learners of English who use American Sign Language. ICICLE utilized a grammar that included mal-rules to allow the parser to generate multiple interpretations for ill-formed input. To select the most plausible parse, the system consulted the existing learner model to determine which parse was more likely.

To mitigate the computational bottlenecks of parsing ill-formed learner productions at runtime, Rudzewitz et al. (2018) proposed a novel architecture that utilized an offline hypothesis generation mechanism. Starting from the well-formed target responses expected for specific exercises, the system pre-computed a constrained search space of both well-formed and ill-formed response variations, storing them alongside their corresponding error diagnoses. At runtime, an online component matched the actual student input to these pre-computed hypotheses.

Fine-grained, pedagogically relevant grammatical error identification, therefore, has often been constrained by either the exercise type or by the

range of grammatical properties covered, as writing mal-rules that can reliably detect the full spectrum of potential learner errors for all grammatical properties in free writing is an unfeasible task.

3 Methodology

The proposed framework for identifying pedagogically relevant grammatical properties consists of a GEC module and a subsequent rule-based error characterization module. In this sense, the framework resembles that of ERRANT, with the key distinction that the grammatical properties characterized are at a finer level of granularity and are designed to align with L2 learning curricula, making them appropriate for learner modeling. To evaluate whether the identified errors are meaningful for the process of language acquisition, we adopt an extrinsic evaluation approach via a downstream task (Resnik and Lin, 2010), whereby the predictiveness of accuracy features extracted based on the identification of errors for an automatic proficiency assessment model is used as a proxy for whether the errors can be used to track language development. By utilizing interpretable machine learning models, the predictive behavior of each feature can be visualized, and a determination about the sensibility of this behavior can be made by human experts.

In the remainder of this section, we first present the rule-based tool developed to identify European Portuguese grammatical properties, dubbed SABER¹. We then describe the learner corpus used in this study and the LLM-powered GEC approach. Next, we describe the adopted text alignment strategy and the accuracy inference algorithm devised for this study. Finally, we describe the experiment used for extrinsic evaluation.

3.1 Grammatical Property Identification

SABER is a rule-based NLP tool designed to identify formal grammatical properties in European Portuguese, which are aligned with the *Referencial Camões*², an authoritative reference mapping grammatical properties to the CEFR levels in which they are taught (A1–C2). SABER facilitates the extraction of 252 grammatical properties, comprising 147 word-level and 105 sentence-level properties.

The tool integrates both Stanza (Qi et al., 2020) and spaCy’s medium Portuguese NLP pipelines

to leverage their respective strengths at annotating the texts with linguistic information. Stanza serves as the primary pipeline due to its superior robustness in Portuguese lemmatization, while spaCy is retained for its specialized capabilities in named-entity recognition and multi-word token processing. This dual pipeline integration allows each pattern matching function, written using spaCy’s token, phrase, and dependency matchers, to utilize the annotations of the pipeline most relevant to the targeted grammatical property. To systematically mitigate upstream NLP annotation errors, SABER employs alternative patterns for each grammatical property designed to handle such preprocessing errors. At the time of development, these identification rules were iteratively tested and fine-tuned using synthetic, property-dense texts generated via GPT-4, optimizing the precision and recall of each rule. To evaluate the tool, five positive and negative test cases were generated for each property using Gemini 3.1, which were subsequently manually validated, resulting in an F₁ score of 85.70%, a recall of 80.48%, and a precision of 96.29%. SABER is available online as a Streamlit application³, and features extracted from an earlier version of it were used in Akef et al. (2025).

Importantly, SABER is designed to identify grammatical properties in well-formed texts, and may not be as robust on ill-formed texts. While this limitation may be considered as a drawback of the tool, it is in fact exploited to characterize learners’ errors in the accuracy inference algorithm described in Section 3.5.

3.2 Dataset

The dataset used to evaluate our framework is the COPLE2 learner corpus (Mendes et al., 2016), which contains 1,634 texts produced by L2 Portuguese learners as part of class examinations at the Instituto de Cultura e Língua Portuguesa (ICLP). The corpus is annotated with error tags and metadata, such as the learner’s L1, task prompt, and fine-grained CEFR levels (A1–C1). These metadata are particularly relevant for the current study, as we aim to test whether LLM prompts providing additional information about the task and the learner’s linguistic background result in corrections that are more informative about the learner’s proficiency.

For this purpose, three additional subsets of the

¹Sistema de Análise e Busca de Estruturas Relevantes

²<https://www.instituto-camoes.pt/activity/centro-virtual/referencial-camoes-ple>

³<https://saber-online.pt>

corpus were extracted for L1-specific proficiency assessment experiments. Based on the availability of texts for each L1, these subsets include L1 Chinese (both Mandarin and Cantonese), L1 Romance (Spanish, Italian, French, and Romanian), and L1 Germanic (English, German, and Dutch). The distribution of the texts in the entire corpus and each subcorpus across the five proficiency levels is presented in Table 1.

Subcorpus	A1	A2	B1	B2	C1
COPLE2	327	419	377	315	196
L1 Chinese	45	104	101	88	100
L1 Romance	171	132	151	117	62
L1 Germanic	71	95	67	60	11

Table 1: Distribution of texts across CEFR levels in COPLE2

3.3 Grammatical Error Correction

Leveraging LLMs’ ability to perform GEC in a zero-shot fashion (Kobayashi et al., 2024; Davis et al., 2024), we performed preliminary experiments by providing a detailed prompt to a range of open-source and proprietary models (including different versions and sizes of Gemma, Llama, Mistral, and Gemini accessed through an inference hosting provider and Google AI Studio) and by examining each model’s corrections of one text from COPLE2. The temperature, a proxy for the degree of creativity in LLMs, was set to the relatively low value of 0.2 (in a range of 0 to 2) for all models, as a measure of preventing over-correction by the model. The most cost-efficient model that satisfied the prompt instructions in the preliminary experiments was Gemini 2.0 Flash, and it is thus the main LLM used in this study, as the objective of this study is not to establish the state of the art in GEC, but rather to demonstrate the feasibility of leveraging corrections produced by even a relatively modest LLM for pedagogically informed learner modeling. For this purpose, we used three prompt variants to generate three corrected versions of each text in COPLE2: a minimalistic prompt (see Appendix A), a detailed prompt including the task instructions, and a detailed prompt including both the task instructions and the learner’s L1 (see Appendix B), to test whether the LLM could leverage this additional information to produce more informative corrections. Additionally, we used the annotator-provided corrections in COPLE2 as a reference for

evaluating the LLM-generated corrections through an extrinsic automatic proficiency assessment task. Moreover, to ensure that this framework is not specific to any one LLM, we utilized corrections made using Gervásio 70B PTPT (Santos et al., 2024), an open-source LLM for European Portuguese, with the detailed prompt including task instructions, as an alternative LLM to compare Gemini’s performance to.

3.4 Text Alignment

The alignment algorithm used is the linguistically enhanced Damerau-Levenshtein algorithm introduced by Felice et al. (2016) and used in ERRANT (Bryant et al., 2017). The key advantage of this alignment algorithm is its consideration for words’ lemmas and parts-of-speech, facilitating alignment in cases of word order errors. As two distinct NLP annotation pipelines are used in SABER, each corrected text also needs to be aligned twice due to the two pipelines using different multi-word tokenization strategies. At the time of error characterization based on the accuracy inference algorithm, each grammatical property utilizes the alignment corresponding to the NLP annotation pipeline used in SABER for that property.

3.5 Accuracy Inference Algorithm

For every property identified in the corrected version, the following scenarios are explored: If the same property is identified in the same aligned span of the learner-produced text, this is considered as a signal that the learner produced the property correctly. There is, however, a caveat: There may be cases where the same property is identified despite the form produced by the learner being different from the corrected form. For instance, the property *Subordinada adverbial (temporal/causal/final)* is matched for both the learner-produced form “*para banharmo*” and the corrected form “*para banhar.*” To account for these inaccuracies and at the same time allow for minor orthographical errors, the two aligned spans must have a normalized Levenshtein edit distance of θ or smaller to be considered as accurate, which is a threshold we determined empirically by comparing the performance of automatic proficiency assessment models trained on features extracted using the annotator-corrected version of the texts and based on different values of θ (see Section 4.1). On the other hand, if the property identified in the corrected version in a particular span is not equivalent to the property identified in the

corresponding span of the learner-produced text (or if no property is identified in the learner-produced text in that span), this is inferred as an inaccuracy, unless the forms of the two spans are identical. This could indeed happen in cases where other tokens that are not within the span of the identified property are different and thus affect the dependency parser’s annotations. In such cases, mismatch between the properties identified is attributed to the NLP pipeline’s error on the learner-produced text and the property identified in the corrected version is marked as accurate.

It is important to note that the final property stored (whether marked as accurate or inaccurate) is the one identified in the corrected version, even when a different property is identified in the same span of the learner-produced version. The rationale for this decision is that the learner’s failure to produce the target property correctly is more informative about the developmental stage of that target property than the inaccurately produced property itself. For instance, if an L2 learner of Portuguese uses the indicative mood instead of the subjunctive mood with *talvez* (maybe), this is a stronger signal that they have not yet (fully) acquired the subjunctive mood than that they have not acquired the indicative mood. In fact, the learner may be fully aware that the indicative mood is inappropriate in that utterance and choose it only to achieve their communicative objective. The full flowchart of the proposed accuracy inference algorithm is presented in Figure 1.

3.6 Evaluation

To perform extrinsic evaluation of whether accuracy features extracted using the accuracy inference algorithm can be used to track L2 development, 28 automatic proficiency assessment models were trained via five-fold cross-validation on various feature sets resulting from different text correction strategies. Additionally, models trained on broad linguistic complexity features, which have been shown to be predictive of learner proficiency (Ribeiro-Flucht et al., 2024), were included to test the contribution of accuracy features extracted in this study against.

3.6.1 Features

Following the characterization of grammatical errors in learners’ texts, accuracy features are calculated by dividing the number of accurate occurrences of a property by all attempts of that property,

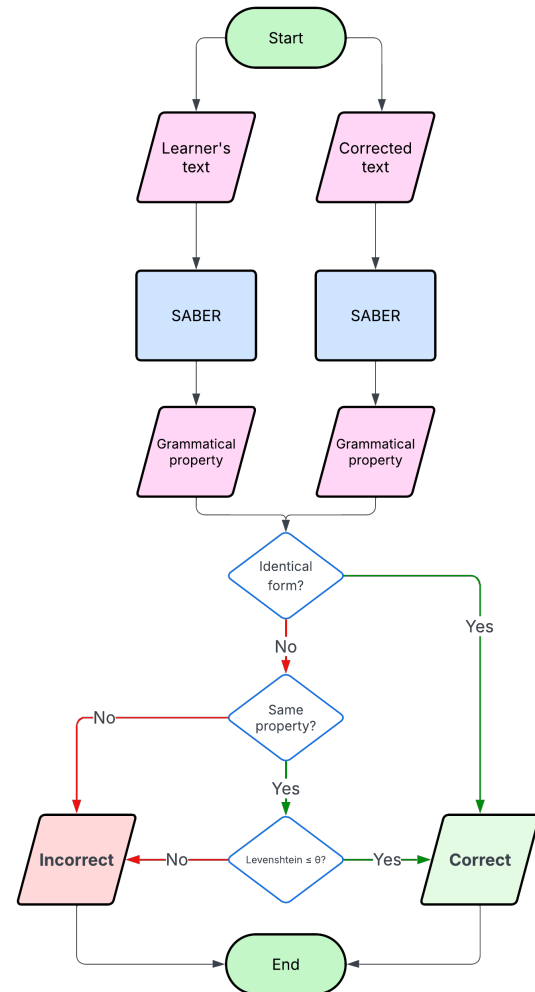


Figure 1: Flowchart representing the property accuracy inference algorithm

represented as the sum of all accurate and inaccurate occurrences, including occurrences of inaccuracies resulting from the omission of the property in a text. Crucially, in cases where a property is appropriately absent from the text (meaning the counts of both accurate and inaccurate occurrences are zero), the value of the accuracy feature for that property is left as a missing value. The rationale for this decision is that not needing a property or successfully avoiding it is a qualitatively different signal from any accuracy value in the range of 0 to 1. Therefore, while the value of the property accuracy feature would be a missing value in such cases, it would be an informative missing value. This is analogous to not having a particular medical test’s results when trying to predict the condition of a patient: The fact that the test result is missing indicates that it was not prescribed by a medical professional, which itself is an informative signal.

This would, however, mean that the machine learning algorithm that is selected must indeed be able to treat missing values as informative. Moreover, as many properties in a text are sparse, the overall accuracy of properties at the same CEFR level were also included as additional features.

The system used to extract broad linguistic complexity features is CTAP (Chen and Meurers, 2016; Weiss and Meurers, 2019) which can extract a total of 489 complexity features for Portuguese and has previously been used for automatic proficiency assessment of Portuguese (Ribeiro-Flucht et al., 2024). The features can be classified into the five classes of count-based, lexical, syntactic, morphological, and discourse features.

3.6.2 Training Algorithm

Aiming for not only robust performance but also innate interpretability, we opted for explainable boosting machines (EBMs; Nori et al., 2019) as the training algorithm used for automatic proficiency assessment. EBMs combine generalized additive models (GAMs) with modern machine learning techniques such as bagging and boosting to achieve performance on par with the highest performing feature-based machine learning models while affording inherent interpretability.

An important advantage of EBMs is that due to the round-robin nature of training, the effect of collinearity is mitigated (Nori et al., 2019). This is especially important in this project, where a wide array of features is utilized, many of which are expected to correlate. Another advantage is EBMs’ handling of missing feature values by treating missing values as a separate value. For the purposes of visualization, EBM’s library recommends that missing features be set to an extreme value (InterpretML Team, 2021). The value of missing accuracy features was thus set to -1. This assignment would not interfere with the model’s prediction for the other values in the range of 0 to 1 due to the EBM’s binning strategy, which places values in bins, with -1 being in its own separate bin. Finally, due to their handling of collinearity, EBMs do not require external feature selection. Caruana (2020) posits that feature selection could result in better performance in EBMs but that the best feature selection strategy is based on the features contributing the most to the prediction of an already trained EBM model using the full feature set.

Threshold	Macro F_1
0.0	52.55% $\pm$ 2.56%
0.05	52.72% $\pm$ 1.63%
0.1	51.77% $\pm$ 2.57%
0.15	52.86 $\pm$ 2.51
0.2	52.62% $\pm$ 1.79%
0.25	52.91% $\pm$ 1.76%
0.5	53.54% $\pm$ 2.41%
1	53.47% $\pm$ 3.19%

Table 2: Performance of EBM models trained on accuracy features extracted using different threshold values based on the annotator-corrected version of COPLE2 texts

4 Results and Discussion

4.1 Orthographical Error Threshold

As introduced in Section 3.5, various values for θ were used to infer grammatical accuracy in learners’ texts. Specifically, the relevant spans share an identical grammatical property with their counterpart in the annotator-corrected version, despite differing in surface form. These values were 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, and 1, where a θ value of 0 represented extreme strictness and a value of 1 represented extreme leniency. The results of automatic proficiency assessment using property-based features extracted using each threshold value are presented in Table 2.

The results indicate that the relatively high threshold of 0.5 results in the features most capable of modeling L2 proficiency. While a normalized Levenshtein distance of 0.5 is unlikely to represent mere orthographical errors in longer spans, the better results achieved from the thresholds of 0.5 and 1 compared to the lower thresholds suggest that as long as the learner has produced the target grammatical property correctly, other lexical (or distinct grammatical) errors made within that production are likely not to undermine the learner’s knowledge of the target property.

4.2 Framework Evaluation

The results of the various models trained via five-fold cross-validation reported according to macro F_1 are presented in Table 3.

It can be observed that accuracy features extracted using the framework described in this paper demonstrate predictiveness of language proficiency with a macro F_1 in the range of 52% to 55%. While

Feature set	General	L1 Chinese	L1 Romance	L1 Germanic
CTAP	58.67 $\pm$ 3.45	63.74 $\pm$ 4.32	50.35 $\pm$ 2.63	50.05 $\pm$ 1.68
Annotator-based	53.47 $\pm$ 2.46	60.55 $\pm$ 3.66	47.39 $\pm$ 4.40	50.71 $\pm$ 6.15
Gemini-based – minimalistic	52.00 $\pm$ 2.26	55.42 $\pm$ 4.68	47.36 $\pm$ 4.60	47.77 $\pm$ 5.39
Gemini-based – detailed + task	53.44 $\pm$ 1.49	57.02 $\pm$ 3.65	45.95 $\pm$ 6.08	48.86 $\pm$ 6.65
Gemini-based – detailed + task + L1	54.92 $\pm$ 1.40	56.11 $\pm$ 2.52	47.49 $\pm$ 6.24	50.02 $\pm$ 5.49
Gervásio-based – detailed + task	53.03 $\pm$ 1.14	57.11 $\pm$ 5.03	44.07 $\pm$ 4.82	47.69 $\pm$ 3.75
CTAP and Gemini-based – detailed + task + L1	59.71 $\pm$ 1.73	63.09 $\pm$ 3.93	50.38 $\pm$ 4.92	51.34 $\pm$ 2.11

Table 3: Performance of EBM models according to macro F_1 trained on different feature sets and corrected subcorpora

CTAP complexity features demonstrate superior predictiveness with a macro F_1 score of 58.67%, we observe an improved performance up to 59.71% with the inclusion of accuracy features.

Moreover, we can observe that using a more detailed prompt, along with providing information about the task, consistently resulted in better performance than the minimalistic prompt, often approaching (and in some cases surpassing) the results based on the annotator’s corrections. Moreover, the inclusion of information about the learner’s L1 resulted in gains compared to cases when this information was excluded, but these cases were not consistent across all L1s, with a slight drop observed from L1 Chinese learners. Nevertheless, these results may be a promising indication of LLMs’ ability to capture traces of cross-linguistic influence in a zero-shot fashion.

Relatedly, while L1-specific Chinese models resulted in consistent improvement over the general model, L1-specific Romance and Germanic models consistently resulted in worse performance. This result may reflect class distribution effects from severe C1 imbalance in the L1 Romance and L1 Germanic datasets. (While the L1 Chinese dataset also has fewer samples at the A1 level, the distinction between learners at lower levels is often easier to model than at more advanced levels, where there is more variability among learners.)

The results obtained from the Gervásio-based features also support the applicability of this framework to other LLMs besides Gemini and supports the choice of Gemini 2.0 Flash as a low-cost but well-performing LLM for this particular task.

To further test the LLM-based accuracy feature extraction approach, we compared the top 30 fea-

tures of the best Gemini-based model with those of the annotator-based model. The overlap was high (29/30), indicating consistent feature prioritization. Because overlap does not guarantee similar informativeness, we compared contribution plots for three shared features: Accuracy of *Pretérito imperfeito do indicativo – verbos regulares*, *Presente do conjuntivo – verbos irregulares*, and *Subordinada adverbial (condicional/concessiva)* (Figures 2, 4, and 6 for Gemini; Figures 3, 5, and 7 for the annotator model).

While differences can be observed between the corresponding graphs, general patterns regarding how the use and accuracy of each feature develops from one proficiency level to the next are similar between the two models. Investigating the cases where noticeable differences exist between the two models (such as the prediction that inaccurate productions of *Presente do conjuntivo – verbos irregulares* are likely to occur at level C1 according to the annotator-based model in Figure 5, but not so according to the Gemini-based model in Figure 4), we observed differences in how the learner’s production was rendered grammatical, an example of which is as follows:

Learner: *Que seja com o telemóvel ou com o computador, podemos ficar em contacto permanente com os amigos ou com a família, assim que temos pouco para dizer uns aos outros.*

Annotator: *Quer seja com o telemóvel ou com o computador, podemos estar em contacto permanente com os amigos ou com a família, pelo que tenhamos pouco para dizer uns aos outros.*

Gemini: *Quer seja com o telemóvel ou com o computador, podemos ficar em contacto permanente com os amigos ou com a família, de tal forma*

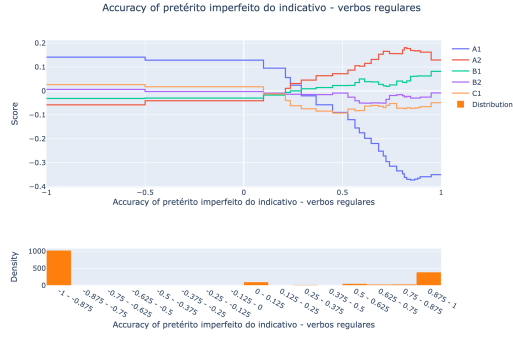


Figure 2: Contribution of Accuracy of *Pretérito imperfeito do indicativo – verbos regulares* to the Gemini-based model

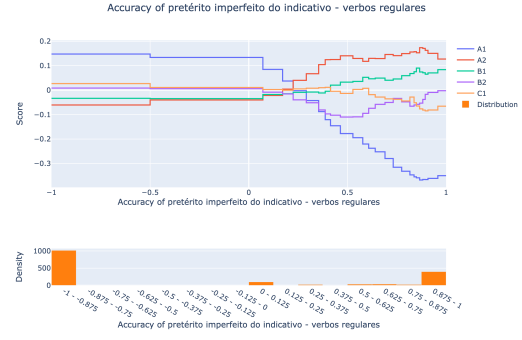


Figure 3: Contribution of Accuracy of *Pretérito imperfeito do indicativo – verbos regulares* to the annotator-based model

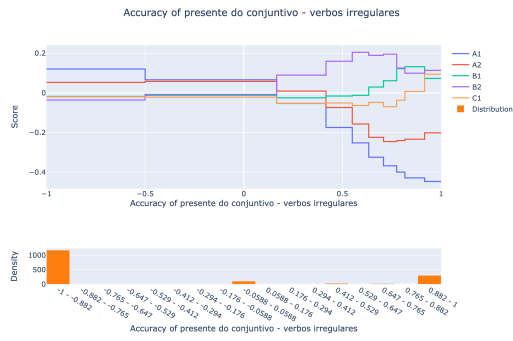


Figure 4: Contribution of Accuracy of *Presente do conjuntivo – verbos irregulares* to the Gemini-based model

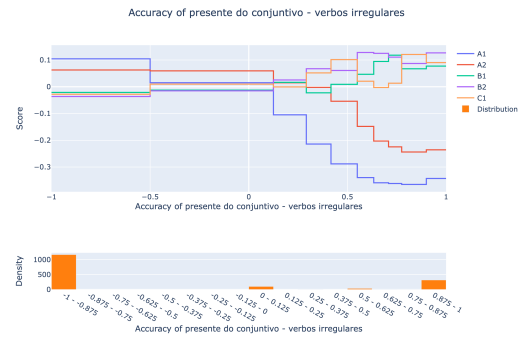


Figure 5: Contribution of Accuracy of *Presente do conjuntivo – verbos irregulares* to the annotator-based model

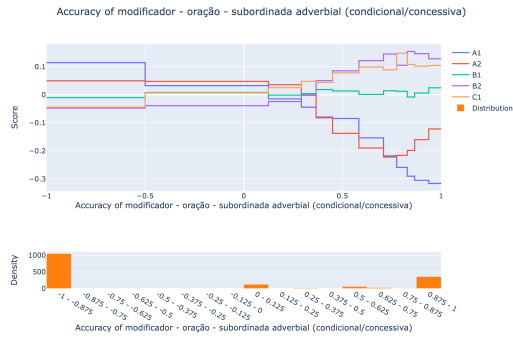


Figure 6: Contribution of Accuracy of *Modificador – oração – subordinada adverbial (condicional/concessiva)* to the Gemini-based model

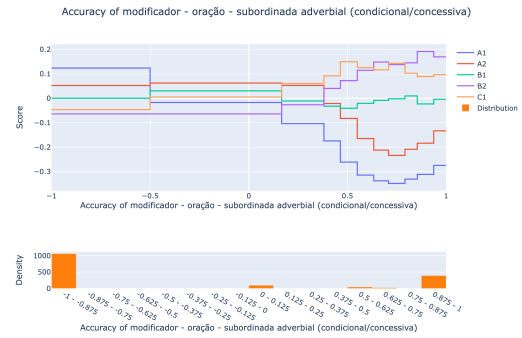


Figure 7: Contribution of Accuracy of *Modificador – oração – subordinada adverbial (condicional/concessiva)* to the annotator-based model

que temos pouco para dizer uns aos outros.

We can observe that while Gemini's correction diverges from that of the annotator in this instance, it is in fact the preferable correction, which could explain why the Gemini-based model with the most detailed prompt including both the task and the learner's L1 outperformed the annotator-based model.

5 Conclusion

To be used as a basis for making determinations about a learner's competencies, GEC must be performed by hypothesizing what the learner intended to produce and what possible factors led them to produce the ill-formed output. While LLMs may be suboptimal for pedagogically-informed grammatical error correction in every context, their capacity to render texts grammatical may still be leveraged for L2 learning applications.

For this purpose, we proposed a framework for characterizing pedagogically relevant grammatical errors corrected by an LLM using a rule-based tool capable of identifying European Portuguese grammatical properties in a fine-grained manner. With the development of similar tools for other languages, such as English (Sagirov and Chen, 2025) and German (Löfflad et al., 2025), the same framework can be applied to learner-produced texts of those languages to construct and update learner models.

Additionally, to evaluate our framework, we opted for an extrinsic evaluation approach by training automatic proficiency assessment models on accuracy features extracted based on either annotator-corrected texts or LLM-corrected texts and demonstrated not only largely similar performance between the two models, but also performance comparable to models trained on linguistic complexity features, which have previously been shown to be highly predictive of proficiency.

This framework enables the construction of highly detailed learner models by allowing learners' free-writing productions to be analyzed and monitored over time, thereby supporting adaptive and personalized language education through ITSs and ICALL systems. Consequently, while more powerful LLMs offer considerable potential for adaptive language education, realizing this potential requires sustained effort to develop NLP tools that pedagogically ground LLM outputs.

Limitations

Owing to the rule-based nature of SABER, it primarily covers formal grammatical properties and only a few use-based features. While a form may be taught at a lower CEFR level, certain uses (or functions) of that same form may correspond to higher CEFR levels, which SABER cannot currently distinguish.

Moreover, because the so-called accuracy features also encode information about the presence or absence of various grammatical properties, much of their predictiveness in the automatic proficiency assessment model is likely attributable to this information, making "accuracy" an operational label rather than a simple binary correctness score.

Furthermore, because the LLMs selected in this study were intended to demonstrate the feasibility of our proposed framework, more powerful reasoning-enabled LLMs could improve both over-

all and L1-specific model performance; however, this possibility was not explored in the present study.

Acknowledgments

This work was developed within the scope of the project *Promoção da Aquisição e ensino do Português como Língua de Herança através de Ferramentas Digitais Inteligentes*, financed by the Foundation for Science and Technology – FCT of the Republic of Portugal and the Camões Institute. We would like to thank anonymous reviewers for their insightful comments on a previous version of this paper.

References

- Soroosh Akef, Detmar Meurers, Amália Mendes, and Patrick Rebuschat. 2025. [Interpretable machine learning for societal language identification: Modeling English and German influences on Portuguese heritage language](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 50–62, Tallinn, Estonia. University of Tartu Library.
- 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.
- 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*, 49(3):643–701.
- Susan Bull. 1994. [Student modelling for second language acquisition](#). *Computers & Education*, 23(1):13–20.

- Rich Caruana. 2020. [Comment on issue #179: Importance of feature selection for EBM](#). GitHub issue comment. Accessed: 2026-02-23.
- Thierry Chanier, Michael Pengelly, Michael Twidale, and John Self. 1992. [Conceptual Modelling in Error Analysis in Computer-Assisted Language Learning Systems](#). In Merryanna L. Swartz and Masoud Yazdani, editors, *Intelligent Tutoring Systems for Foreign Language Learning*, pages 125–150. Springer Berlin Heidelberg, Berlin, Heidelberg.
- 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 (CL4LC)*, pages 113–119, Osaka, Japan. The COLING 2016 Organizing Committee.
- 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.
- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipour, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Trude Heift. 2005. [Inspectable learner reports for web-based language learning](#). *ReCALL*, 17(1):32–46.
- Trude Heift. 2008. [Modeling learner variability in CALL](#). *Computer Assisted Language Learning*, 21(4):305–321.
- Trude Heift and Paul McFetridge. 1999. [Exploiting the Student Model to Emphasize Language Pedagogy in Natural Language Processing](#). In *Computer Mediated Language Assessment and Evaluation in Natural Language Processing*.
- InterpretML Team. 2021. [Comment on issue #18: Doing some treatment for missing values](#). GitHub issue comment. Accessed: 2026-02-23.
- 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. [Large Language Models Are State-of-the-Art Evaluator for Grammatical Error Correction](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.
- Denise Löfflad, Benedikt Beuttler, and Detmar Meurers. 2025. [German Grammar Profile for Learners: Pedagogical Feature Definition and Automated Extraction](#). In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, pages 212–223, Hannover, Germany. HsH Applied Academics.
- 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.
- Arianna Masciolini, Andrew Caines, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025. [The MultiGEC-2025 Shared Task on Multilingual Grammatical Error Correction at NLP4CALL](#).
- Amália Mendes, Sandra Antunes, Maarten Janssen, and Anabela Gonçalves. 2016. [The COPLE2 corpus: a learner corpus for Portuguese](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 3207–3214, Portorož, Slovenia. European Language Resources Association (ELRA).
- Lisa N. Michaud, Kathleen F. McCoy, and Rashida Z. Davis. 2005. [A Model to Disambiguate Natural Language Parses on the Basis of User Language Proficiency: Design and Evaluation](#). *User Modeling and User-Adapted Interaction*, 15(1):55–84.
- 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 Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.
- Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. [InterpretML: A Unified Framework for Machine Learning Interpretability](#). *arXiv preprint*. ArXiv:1909.09223 [cs].
- 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 [cs].

- Philip Resnik and Jimmy Lin. 2010. [Evaluation of NLP Systems](#). In *The Handbook of Computational Linguistics and Natural Language Processing*, pages 271–295. John Wiley & Sons, Ltd.
- Luisa Ribeiro-Flucht, Xiaobin Chen, and Detmar Meurers. 2024. [Explainable AI in language learning: Linking empirical evidence and theoretical concepts in proficiency and readability modeling of Portuguese](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 199–209, Mexico City, Mexico. Association for Computational Linguistics.
- Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. 2018. [Generating Feedback for English Foreign Language Exercises](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–136, New Orleans, Louisiana. Association for Computational Linguistics.
- Nelly Sagirov and Xiaobin Chen. 2025. [POLKE: A system for comprehensively annotating pedagogically-oriented grammatical structure use in language production](#). *Manuscript submitted for publication to Behavior Research Methods*.
- Rodrigo Santos, João Silva, Luís Gomes, João Rodrigues, and António Branco. 2024. [Advancing generative AI for Portuguese with Open Decoder Gervásio PT-*](#). *Preprint*, arXiv:2402.18766.
- David Schneider and Kathleen F. McCoy. 1998. [Recognizing Syntactic Errors in the Writing of Second Language Learners](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1198–1204, Montreal, Quebec, Canada. Association for Computational Linguistics.
- Zarah Weiss and Detmar Meurers. 2019. [Analyzing linguistic complexity and accuracy in academic language development of German across elementary and secondary school](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 380–393, Florence, Italy. Association for Computational Linguistics.
- Jing Xu and Susan Bull. 2010. [Encouraging advanced second language speakers to recognise their language difficulties: a personalised computer-based approach](#). *Computer Assisted Language Learning*, 23(2):111–127.
- Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwar, 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.
- Robert Östling, Murathan Kurfali, and Andrew Caines. 2025. [LLM-based post-editing as reference-free GEC evaluation](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 213–224, Vienna, Austria. Association for Computational Linguistics.

A Minimalistic Prompt

Correct the following text according to standard European Portuguese grammar. Output only the corrected text.

B Detailed Prompt

You are an expert proofreader and teacher of European Portuguese, specializing in correcting texts written by language learners. Your primary goal is to correct the "Learner's Text" you will be given.

Adhere strictly to these instructions:

1. ****Correction Scope:**** Focus on fixing clear errors in European Portuguese grammar, spelling, punctuation, and word choice.
2. ****Minimal Edits:**** Implement the fewest changes necessary. Preserve the learner's original meaning, voice, and sentence structure as much as possible. Only alter what is incorrect or hinders comprehension. Do not rephrase for stylistic preference if the original is grammatically acceptable. That being said, cases that would still be acceptable in Brazilian Portuguese but not so in European Portuguese must also be corrected. Also, you must take the whole sentence into account for grammaticality. If you need to change the word form the learner produced to make the corrected sentence grammatical, you are allowed to do it.
3. ****Intent and Context:**** The learner wrote their text in response to a specific "Task Prompt". You will receive this "Task Prompt" along with the "Learner's Text". Interpret and correct errors based on what the learner likely intended to communicate to fulfill that "Task Prompt".
4. ****L1 Interference (Negative Transfer):**** You will be provided with the learner's native language (L1). Actively utilize this information to identify instances of negative transfer. When determining the correct form, base your decision on what the learner likely intended to produce given typical L1 interference patterns.
5. ****Language Variant:**** All corrections must use standard European Portuguese, not Brazilian Portuguese.
6. ****Output Format:**** Your response **MUST BE ONLY** the fully corrected European Portuguese

text. Do NOT include any explanations, apologies, greetings, or any other text before or after the corrected version.

You will receive the "Learner's L1", the "Task Prompt", and the "Learner's Text" clearly labeled in the user message.

Teaching Through Analogies: A Modular Pipeline for Educational Analogy Generation

Mariam Barakat¹ and Ekaterina Kochmar^{1,2}

¹Mohamed bin Zayed University of Artificial Intelligence, ²University of Victoria
 {mariam.barakat, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

Analogies help learners understand unfamiliar concepts by relating them to known concepts. Despite recent advances, large language models (LLMs) continue to struggle to generate analogies of comparable quality to those produced by humans. We present a modular pipeline for educational analogy generation, decomposing the task into four stages: source finding, sub-concept generation, explanation generation, and evaluation. Grounded in Structure Mapping Theory, the pipeline enables systematic, stage-by-stage analysis of how model choice and input configuration affect analogy quality. We evaluate 12 state-of-the-art LLMs across six model families on two datasets with structured sub-concept annotations (SCAR and ParallelPARC), alongside seven embedding models for closed-setting retrieval. Our results show that sub-concepts substantially improve explanation quality and closed setting retrieval precision but provide limited benefit in open-ended source generation. We further introduce an LLM-as-a-judge evaluation methodology and validate its scoring against human annotations from seven annotators, finding that Claude Sonnet 4.6 aligns more reliably with human rankings than with fine-grained absolute scores. Taken together, our findings reveal cross-stage interactions that isolated studies cannot capture, and highlight sub-concept grounding as a key driver of analogy quality generation.

1 Introduction

As educational systems evolve in the age of AI, memorisation is no longer sufficient. Instead, students are required to be able to reason across domains and understand underlying structures – skills that are foundational to critical thinking in a rapidly changing world. One of the effective ways to develop these abilities is through analogies (Eriksson et al., 2024), which facilitate learning by mapping unfamiliar knowledge (*target system*) to prior

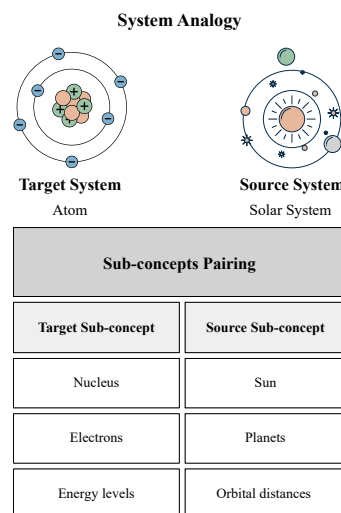


Figure 1: Illustrative example of a system analogy between the *atom* and the *solar system*, showing the source and target systems and their aligned sub-concepts.

knowledge (*source system*) through relational structures (Ye et al., 2024; Czinczoll et al., 2022; Bettin and Ott, 2023). Even when a significant cognitive gap exists, intermediate “bridging” analogies can guide students toward comprehension (Brown and Clement, 1989). In this work, we frame analogy generation as a modular pipeline to enable systematic analysis and improvement of each stage of the process.

Following previous research (Yuan et al., 2023a), we use the term *system* rather than *concept* to emphasise that analogies map multiple components. We refer to these constituent properties as *sub-concepts* throughout this paper, using “sub-” to emphasise that they are components of the larger system rather than standalone concepts.

As shown in Figure 1, explaining the *atom* (target system) through the *solar system* (source system) requires the explained mapping of individual sub-concepts: the *nucleus* maps to the *sun*, *electrons* map to *planets*, and *energy levels* map to

orbital distances.

Automatically generating high-quality analogies remains an open challenge, as poorly constructed analogies can introduce or reinforce misconceptions through incorrect mappings or oversimplification of underlying concepts (Zhou et al., 2025; Shao et al., 2025).

Despite recent progress, analogy generation research remains fragmented: prior work addresses retrieval, generation, or evaluation in isolation, without connecting them into a unified pipeline (Yuan et al., 2023a,b; Zhai et al., 2024; Sultan et al., 2024). Source finding relies predominantly on embedding-based retrieval over closed candidate pools, favouring surface similarity over structural alignment (Yuan et al., 2023a; Czinczoll et al., 2022; Jiayang et al., 2023), while sub-concept annotations in SCAR (Yuan et al., 2023a) and ParallelPARC (Sultan et al., 2024) are used for dataset construction rather than to guide generation. Evaluation remains under-addressed, as neither lexical metrics nor LLM-as-a-judge approaches align reliably with human judgments (Zhai et al., 2024; Bhavya et al., 2024), and most studies evaluate GPT-4 variants exclusively (Yuan et al., 2023a; Kim et al., 2023; Sultan et al., 2024; Shao et al., 2025; Li et al., 2024), leaving cross-model generalisability unexplored.

In this paper, we make three contributions: (1) a modular pipeline for educational analogy generation, grounded in Structure Mapping Theory (Gentner, 1981), decomposed into source finding, sub-concept generation, explanation generation, and evaluation; (2) a systematic cross-model evaluation across 12 LLMs from six families, finding that sub-concepts improve retrieval and explanation quality but offer limited benefit in open-ended source generation; and (3) a closed-to-open evaluation methodology, moving from controlled evaluation against gold annotations on SCAR (Yuan et al., 2023a) and ParallelPARC (Sultan et al., 2024) to LLM-generated candidates with baseline, embedding-based, and re-ranking selection strategies.

Each stage in our proposed pipeline corresponds to a research question:

RQ1: Source Finding. Can LLMs identify suitable source systems for a given target in both closed and open settings, moving beyond predefined candidate pools?

RQ2: Sub-concept Generation. Can LLMs reliably generate sub-concept mappings given a target-

source pair, and at which pipeline stages do sub-concepts improve performance?

RQ3: Explanation Generation. How well can LLMs produce full analogical explanations, and what input configurations yield the highest quality?

RQ4: Evaluation. How well does LLM-as-a-judge evaluation align with human judgments of analogy quality? Moreover, which generation model is the best at source finding in open settings?

All code and data are publicly available at: <https://github.com/Myriam2002/Toward-Usable-Scientific-Analogies>.

2 Related Work

Source Finding. Prior approaches to source finding rely primarily on embedding-based retrieval within closed candidate pools. Yuan et al. (2023a) use greedy and Kuhn-Munkres matching but focus on mapping given concepts rather than discovering new sources. Similarly, Yuan et al. (2023b) employ embedding similarity, but focus only on simple one- or two-relation analogies. Jiayang et al. (2023) show that LLMs are easily distracted by surface similarity, achieving only 30% accuracy compared to 85% for humans, while Ye et al. (2024) report substantial performance gaps on longer analogies. Li et al. (2024) further show that free generation outperforms embedding-based retrieval for analogies in the history domain, suggesting fundamental limitations of similarity-based methods. However, open-ended source discovery, where models must identify suitable sources without predefined candidates, remains largely unexplored.

Sub-concept Generation. Datasets such as SCAR (Yuan et al., 2023a) and ParallelPARC (Sultan et al., 2024) provide rich sub-concept annotations linking source and target systems. However, these annotations are primarily used for dataset construction rather than as inputs to generation models. Yuan et al. (2023a) show that sub-concept matching is critical for analogy quality, yet models achieve only 62% accuracy compared to 86% for humans, even when sub-concepts on both sides are provided. Moreover, evaluation is restricted to closed settings with fully specified inputs. The ability of LLMs to generate sub-concepts in open settings and their impact across pipeline stages has not been studied.

Explanation Generation. A well-matched source and accurate sub-concept mappings are necessary but not sufficient, as the explanation

is what ultimately determines whether a learner grasps the analogy. Producing explanations that capture *why* an analogy holds remains particularly challenging. Zhai et al. (2024) separate concept generation and explanation generation, finding that while LLMs approach human performance on concept generation, they struggle to produce explanations with sufficient relational depth. Bhavya et al. (2024) similarly report that model-generated explanations often lack coherence or drift from the intended analogy. Overall, explanation generation remains underexplored, particularly regarding how input representations affect quality.

Evaluation. Evaluating analogy quality remains challenging. Lexical metrics such as ROUGE and BLEURT fail to capture relational soundness (Zhai et al., 2024), while more structured evaluation frameworks still fall short of fully capturing analogy quality (Bhavya et al., 2024). Human evaluation, although more reliable, is costly (Jiayang et al., 2023; Czinczoll et al., 2022). LLM-as-a-judge methods offer a scalable alternative (Zhai et al., 2024), but their alignment with human judgments in analogy-specific settings remains mediocre.

3 Methodology

We present a systematic evaluation of the proposed modular pipeline for educational analogy generation, analysing how model choice and input configuration influence performance at each stage. The pipeline is decomposed into four stages aligned with our research questions: source finding (RQ1), sub-concept generation (RQ2), explanation generation (RQ3), and evaluation (RQ4). Figure 2 provides an overview of the experimental setup. To illustrate the full pipeline, consider the example in Figure 1, where the target system *atom* is explained using the source *solar system*. In the source-finding stage, candidate source systems are retrieved based on their similarity to the target. Next, sub-concept generation aligns components such as *nucleus* → *sun* and *electrons* → *planets*. The explanation generation stage then produces a natural-language explanation linking these correspondences. Finally, the analogy is evaluated based on coherence, mapping soundness, and explanatory power. This example demonstrates how each stage contributes to the final analogy.

3.1 Datasets

We use two datasets with explicit sub-concept annotations, enabling controlled evaluation across pipeline stages.¹

SCAR (Yuan et al., 2023a) comprises 400 system-level analogy instances across 13 scientific domains, created by human annotators. Each instance pairs a target system with a source system and is annotated with background descriptions, multiple sub-concept mappings, and explanations for these mappings (1,614 total annotated mappings). A representative example is provided in Appendix A.

ParallelPARC (Sultan et al., 2024) extends the ProPara dataset (Tandon et al., 2018) by transforming scientific-process descriptions into analogy pairs through a pipeline of LLM generation, automatic labelling, and human validation. It introduces 310 gold analogies, each annotated with relational alignments between corresponding properties in the format (entity, verb, entity) mapped to (entity, verb, entity). Please refer to Appendix A for an example.

3.2 Models

Closed Setting. For source retrieval in the closed setting, we evaluate a diverse set of embedding models: OpenAI’s embedding-small and embedding-large,² E5-large,³ BGE-large,⁴ MPNet-base,⁵ MiniLM-L6,⁶ and SPECTER.⁷ These models cover complementary paradigms: proprietary general-purpose embeddings (OpenAI), open-source retrieval-oriented models (E5-large, BGE-large), sentence-similarity baselines (MPNet-base, MiniLM-L6), and a domain-specialised scientific encoder (SPECTER), enabling comparison across models that differ in scale, training objective, and domain specificity.

Open Setting. For open-ended generation across all three stages, we use the DSPy framework (Khat-tab et al., 2022, 2024) for consistent input/output

¹Throughout the paper, we adopt the field names from SCAR for clarity, and align corresponding fields from other datasets to match this structure.

²<https://platform.openai.com/docs/guides/embeddings>

³<https://huggingface.co/intfloat/e5-large>

⁴<https://huggingface.co/BAAI/bge-large-en>

⁵<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁶<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

⁷<https://huggingface.co/allenai/specter>

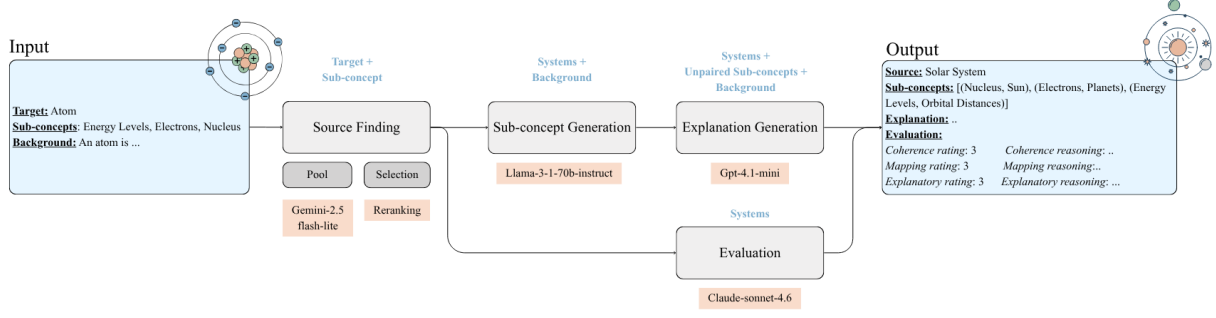


Figure 2: Overview of the proposed modular pipeline for educational analogy generation. Given a target system with its sub-concepts and background description, the framework proceeds through four stages: (1) source finding, where a pool of candidate sources is generated and selected; (2) sub-concept generation; (3) explanation generation; and (4) evaluation using an external LLM-as-a-judge. Each stage operates under input configurations tailored to its requirements, as determined by our empirical results. The output illustrates the final generated analogy, including aligned sub-concepts, explanation, and evaluation scores.

interfaces. We evaluate 12 LLMs across six families: OpenAI’s GPT-4.1 (mini, nano), GPT-OSS (20B, 120B), Meta’s LLaMA 3.1 (8B, 70B, 405B) Instruct, Google’s Gemini 2.5 Flash-Lite, xAI’s Grok-4-Fast, DeepSeek-R1, and Alibaba’s Qwen3 (14B, 32B). Models were selected based on OpenRouter’s public rankings in academia and science,⁸ balancing benchmark performance and cost-efficiency. All models were run at temperature 0.2 for consistency. Models were accessed via OpenAI, OpenRouter, and DeepInfra APIs.⁹

3.3 Source Finding (RQ1): Pool & Selection

Closed Setting: Embedding-based Retrieval.

Following prior work (Czinczoll et al., 2022; Yuan et al., 2023a), we embed all unique source systems from SCAR and ParallelPARC using the seven models from Section 3.2 and retrieve the top-20 candidates per target via cosine similarity. We test four input configurations with increasing context: (1) **target only** to assess surface-level retrieval; (2) **+background** to test whether definitional context helps; (3) **+sub-concepts** to assess the role of structural alignment; and (4) **sub-concepts & background** to evaluate the effect of combining both.

To go beyond similarity-based ranking, we introduce an LLM reranking stage: the top-10 candidates from the best-performing embedding configuration are passed to LLaMA-3.1-70B-Instruct, which selects the top-3 (Appendix B). We evaluate five reranking configurations: (1) **target only** as a baseline; (2) **+background** to test definitional

context; (3) **+gold sub-concepts** to assess structural alignment with gold annotations; (4) **+sub-concepts & background** to combine both signals; and (5) **+generated sub-concepts** to test whether generated sub-concepts can substitute for gold annotations in downstream selection (see Subsection 3.4).

We evaluate retrieval using $Hit@K$ (Eq. 1), which measures whether at least one gold source appears in the top- K results. This evaluation directly measures the quality of the retrieval component by assessing whether embedding models successfully retrieve appropriate source systems for a given target.

$$Hit@K = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\text{rank}_i \leq K) \quad (1)$$

Open Setting: LLM-based Generation. In the open setting, each of the 12 LLMs generates 20 ranked candidate sources from scratch, evaluated on SCAR only due to its educational relevance and computational constraints. Following closed-setting results, we test two input configurations: (1) **target only** and (2) **+sub-concepts**.

Since multiple candidates are generated, we evaluate three selection strategies: (1) **baseline top-1**, selecting the first output from an LLM; (2) **embedding-based top-1**, selecting the candidate with the highest cosine similarity to the target using text-embedding-3-small, strong in closed-setting retrieval and cost-efficient; and (3) **re-ranker top-1**, applying the LLM-based reranking framework from Subsection 3.3 with the **name & subconcepts** configuration, selected based on its strong performance in the closed setting.

⁸<https://openrouter.ai/rankings>

⁹OpenAI: <https://openai.com>, OpenRouter: <https://openrouter.ai>, DeepInfra: <https://deepinfra.com>

We report both *exact Hit@K* (Eq. 1) and *semantic Hit@K*. A generated source is considered correct if its embedding similarity to any gold source exceeds 0.374, corresponding to the upper tertile of pairwise gold similarities in SCAR (see Appendix C for more details). To avoid evaluation bias, we use all-MiniLM-L6-v2 for semantic Hit@K scoring rather than text-embedding-3-small, which is used in the embedding-based Top-1 selection step. Using the same model for both selection and evaluation might unfairly favour that configuration over others.

3.4 Sub-concept Generation (RQ2)

Before tackling the harder generation task, we revisit sub-concept matching – where models align given sub-concepts rather than produce new ones – to establish how well current LLMs capture structural correspondences and how far the field has advanced since Yuan et al. (2023a). Figure 3 highlights the distinction between the two tasks.

Inputs	Sub-concept Matching	Sub-concept Generation
Target System	Atom	
Target Sub-concept	[Energy levels, Electrons, Nucleus]	[Energy levels, Electrons, Nucleus]
Source System	Solar System	
Source Sub-concept	[Planets, Orbital Distance, Sun]	?
Output	[(Nucleus, Sun), (Electrons, Planets), (Energy Levels, Orbital Distances)]	[(Nucleus, Sun), (Electrons, Planets), (Energy Levels, Orbital Distances)]

Figure 3: Sub-concept matching and generation stage (RQ2). Given a target system (e.g., *atom*) and a source system (e.g., *solar system*), the model performs sub-concept matching using gold sub-concepts, and sub-concept generation when source sub-concepts are unavailable, producing aligned sub-concept pairs.

Sub-concept Matching. We re-evaluate sub-concept matching on SCAR, using its reported system accuracy (62% for GPT-4 vs. 86% for human annotators) as a reference point. Models receive randomly shuffled lists of target and source sub-concepts and must identify the correct correspondences. Then, we test two scenarios: (1) **without background**, where models rely only on sub-concept labels to recognise structural alignments; and (2) **with background** (see Listing 1 in Appendix A). We compare performance across the 12 LLMs to determine if the gap between model and human performance can be narrowed. For evaluation, we use *System Accuracy* (Yuan et al., 2023a), which assigns 1 if all pairs are correct and 0 otherwise.

wise.

Sub-concept Generation. To test whether LLMs can generate source sub-concepts in open settings, we test two configurations: given the target system, its sub-concepts, and the source system name, models must produce the corresponding source sub-concepts either (1) **without background** or (2) **with background**.

To evaluate, we use *Semantic Match Accuracy* (SMA). It uses cosine similarity between embeddings built with all-MiniLM-L6-v2.¹⁰ A fixed 0.7 threshold is adopted as a conservative heuristic for MiniLM embeddings, balancing semantic sensitivity with false-positive robustness.

$$\text{SMA} = \frac{\text{count}(\text{similarity} \geq 0.7)}{\text{total concepts}} \quad (2)$$

3.5 Explanation Generation (RQ3)

The third stage evaluates whether LLMs can articulate why an analogy holds, given a target and source. Each model generates a natural-language explanation. To isolate the contribution of different input signals, we test six configurations: (S1) **names only**, (S2) **+background**, (S3) **+unpaired sub-concepts**, (S4) **+unpaired sub-concepts & background**, (S5) **+paired mappings**, and (S6) **+paired mappings & background** (see the prompts in Appendix D). Throughout the results, we refer to these configurations as S1–S6 for brevity.

We evaluate generated explanations against gold references using cosine similarity over all-mpnet-base-v2 (Reimers and Gurevych, 2019) embeddings known for their ability to compare sentences to each other, capturing semantic equivalence beyond surface variation, and then we compare performance across input settings.

3.6 Evaluation (RQ4)

LLM-as-a-judge. To close the pipeline, we turn to evaluation – employing Claude Sonnet 4.6¹¹ as an external LLM judge, independent of all 12 pipeline models to reduce evaluation bias. Analogies are scored on three dimensions (using 1–3 scale): **coherence** (does the pairing make intuitive sense?), **mapping soundness** (are source–target

¹⁰all-MiniLM-L6-v2:<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

¹¹<https://openrouter.ai/anthropic/claude-sonnet-4.6>

correspondences structurally valid?), and **explanatory power** (does the analogy help a learner understand the target?). We use few-shot prompting with 10 calibrated examples spanning the full score range, as preliminary experiments without examples yielded less stable judgments. The full prompt is provided in Appendix E. We apply this judge across all 12 models to identify which one produces the highest-quality analogies overall.

Human Evaluation. To validate the LLM-based ratings, we collect human judgments from seven graduate students in AI and Business.¹² Prior to annotation, participants were provided with detailed instructions describing the task, including definitions of the three evaluation dimensions (coherence, mapping soundness, and explanatory power), along with illustrative examples to calibrate their understanding. Annotators evaluate 45 analogy pairs – 15 target concepts across five domains (check Appendix J for all analogies tested), each paired with three candidate sources – scoring each on the same three dimensions used by the LLM judge. In addition to scoring individual dimensions, annotators rank candidate analogies based on overall learning usefulness, capturing whether the analogy is intuitive and helpful for understanding the target concept. This ranking provides a qualitative, learner-oriented perspective that complements the dimension-level scores, yielding a total of 165 annotated datapoints.¹³

We measure inter-annotator reliability using Krippendorff’s α (Krippendorff, 2011) for ordinal ratings, Kendall’s W (Kendall, 1990) for ranking concordance across annotators, and Spearman’s ρ to compare human scores against LLM judge scores.

4 Results and Analysis

4.1 Source Finding (RQ1)

Closed Setting. Embedding-based retrieval shows consistent but dataset-dependent trends. On SCAR, adding sub-concepts yields the largest gains: OpenAI-Large improves from 0.710 to 0.792 and OpenAI-Small from 0.677 to 0.780, with the average increase from 0.557 to 0.673. Adding background provides no

benefit (avg. 0.629), likely because sub-concepts already capture structural alignment, and extra context introduces noise. On ParallelPARC, performance is near-saturated across settings (name-only avg. 0.876), with sub-concepts + background performing best (OpenAI-Large: 0.942, OpenAI-Small: 0.932, avg. 0.908), as its backgrounds provide richer relational signal than SCAR descriptions. Across both datasets, OpenAI embeddings consistently lead, while SPECTER lags (SCAR avg. 0.437; ParallelPARC avg. 0.781), reflecting weaker alignment with the task. Full results are in Appendix I.

For selection, we compare five configurations: re-ranking with names only, background, gold sub-concepts, gold sub-concepts + background, generated sub-concepts, and embedding-only retrieval. Re-ranking with gold sub-concepts achieves the best Hit@1 (24.5%), followed by background (22.2%), gold + background (20.8%), embedding-only (19.5%), generated sub-concepts (18.2%), and names only (17.5%). At Hit@2 and Hit@3, embedding-only performs best (30.5%, 40.0%), while gold sub-concepts rank third (31.8%, 35.8%), indicating that embeddings better retrieve candidates, whereas LLM re-ranking excels at top-1 selection. Generated sub-concepts outperform the name-only baseline (18.2% vs 17.5% at Hit@1), showing that the pipeline produces meaningful structure without gold annotations, though a gap remains. Overall, embeddings narrow the search space, while LLM reasoning over sub-concepts is key for precise selection.

Open Setting. Adding sub-concepts to the generation prompt yields only marginal gains (avg. 0.4–1.6 pp across all K and models), suggesting limited benefit compared to their substantial impact in retrieval, while incurring additional prompt cost, making target-only prompting the more practical default. The two exceptions are LLaMA-405B and Qwen3-32B, which show a slight decrease when sub-concepts are added. In absolute terms, the strongest Hit@20 results are achieved under the target with sub-concepts mode, with Gemini-2.5 (75.4%) and GPT-OSS-120B (75.3%) leading. However, top models cluster within a few percentage points of each other, suggesting that scale alone does not determine generation quality. Per-model results are in Appendix H.

¹²Annotation form: <https://myriam2002.github.io/analogy-annotation/>

¹³Confidence scores were collected for ranking but not used in the current analysis; incorporating them to weigh annotator reliability is left for future work.

Model	Matching		Generation	
	+BG	−BG	+BG	−BG
Grok-4	0.76	0.78	0.42	0.31
DeepSeek-R1	0.70	0.76	0.45	0.33
GPT-OSS-120B	0.69	0.72	0.34	0.27
Qwen3-14B	0.68	0.70	0.42	0.31
GPT-OSS-20B	0.66	0.68	0.39	0.28
GPT-4.1-mini	0.66	0.68	0.42	0.31
Qwen3-32B	0.65	0.71	0.40	0.31
Llama-70B	0.57	0.60	0.47	0.32
Llama-405B	0.56	0.62	0.47	0.33
Gemini-2.5	0.54	0.64	0.42	0.29
Llama-8B	0.40	0.46	0.44	0.27
GPT-4.1-nano	0.37	0.46	0.34	0.27

Table 1: System accuracy (Matching) and semantic match accuracy (Generation) with (+BG) and without (−BG) background descriptions. Bold indicates the best score per column. Gray shading indicates the higher-scoring mode per task for each model. Models are sorted by matching accuracy.

4.2 Sub-concept Generation (RQ2)

Table 1 contrasts sub-concept matching and generation across models. For matching, performance is slightly higher *without* background descriptions (avg. 0.650 vs. 0.603), suggesting additional context hinders alignment. Grok-4 leads at 0.78, surpassing the prior state-of-the-art of 0.62 (Yuan et al., 2023a), with gains attributed to model choice and prompt design (Appendix F). GPT-4.1-nano performs worst (0.37) despite not being the smallest model. These results motivate the harder generation setting explored next.

For generation, background descriptions provide a clear benefit, raising average semantic match accuracy from 0.298 to 0.412. The LLaMA family leads, with LLaMA-3.1-70B and LLaMA-3.1-405B both reaching 0.47, while GPT-based variants consistently occupy the lower half. Notably, GPT-4.1-nano and GPT-OSS-120B (0.34) underperform the smaller GPT-OSS-20B (0.39), confirming that scale alone does not guarantee stronger generation.

Based on these findings, we adopt LLaMA-3.1-70B with background descriptions for all subsequent generation experiments.

4.3 Explanation generation (RQ3)

The most apparent pattern is the large jump in similarity when sub-concepts are introduced: average scores rise from 0.724 in S1–S2 to 0.854 with unpaired sub-concepts (S3–S4) and 0.830 with paired sub-concepts (S5–S6). By contrast, adding back-

ground descriptions has a negligible effect across all three setting pairs (at most +0.006), consistent with our earlier matching findings. The best setting overall is S5 (systems + paired sub-concepts).

GPT-4.1-Mini is the strongest model overall (mean 0.827) and is the best performer in four of the six settings (S3–S6), indicating strong generalisation across sub-concept conditions. In the system-only settings (S1–S2), where scores are uniformly low and differences between models are small (range ≈ 0.03), no single model stands out, with GPT-OSS-120B and LLaMA-70B marginally leading in S1 and S2, respectively. Check Appendix G for the detailed heatmap.

4.4 Evaluation (RQ4)

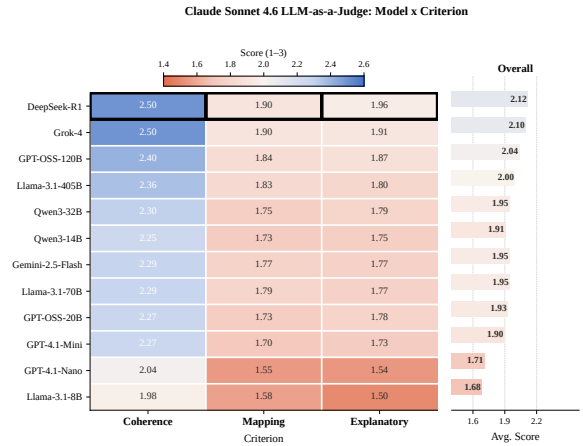


Figure 4: Per-criterion scores assigned by Claude Sonnet 4.6 acting as an LLM judge. Scores are on a 1–3 scale across three criteria: Coherence, Mapping, and Explanatory power. Black borders indicate the top-performing model per criterion.

LLM-as-a-judge. Figure 4 shows LLM-as-a-judge scores across generation models. DeepSeek-R1 and Grok-4 achieve the highest overall performance (2.12 and 2.10), while GPT-4.1-Nano and LLaMA-3.1-8B perform the worst (1.71 and 1.68). The remaining models cluster closely in the middle, indicating relatively small differences across most systems. Across all models, coherence scores are consistently higher than mapping and explanatory scores, suggesting that models often produce intuitively plausible analogies even when structural mappings are incomplete or explanations are less informative.

Table 2 compares selection strategies under LLM-as-a-judge evaluation. The baseline and re-ranking approaches achieve close average scores

Dimension	Baseline		Embeddings		Reranker	
	mean	std	mean	std	mean	std
Coherence	2.463	0.652	1.956	0.748	2.437	0.677
Mapping	1.839	0.560	1.550	0.583	1.878	0.561
Explanatory	1.902	0.622	1.484	0.581	1.894	0.640
Avg. Score	2.068	0.541	1.663	0.575	2.069	0.550

Table 2: Comparison of analogy quality across selection strategies using LLM-as-a-judge scores (1–3 scale).

(2.068 vs. 2.069), but differ in their strengths: the baseline yields slightly higher coherence, while re-ranking improves mapping quality. This indicates that LLM-based selection refines structural alignment without substantially affecting overall performance.

Human Evaluation. Inter-annotator agreement is measured using Krippendorff’s α . Agreement is fair for *coherence* (0.382) and *explanatory power* (0.381), and lower for *mapping soundness* (0.342) – the most technically demanding dimension. Agreement is highest for the overall ranking (0.479) – where annotators rank the three candidate sources per target by overall learning usefulness – reflecting that relative ordering is easier to agree on than assigning consistent absolute scores across all three dimensions; for instance, two annotators may agree that source A is best, yet still differ on whether to rate its coherence a 2 or a 3. Appendix K provides detailed ranking statistics.

Spearman correlation (ρ) between Claude and human annotators varies across both annotators and dimensions. Agreement is highest for *explanatory power* (mean $\rho = 0.511$, range 0.269–0.678) and lowest for *mapping soundness* (mean $\rho = 0.342$, range 0.124–0.507), consistent with LLM-as-a-judge trends. Coherence falls between these extremes (mean $\rho = 0.368$, range 0.130–0.581), reflecting moderate agreement on whether analogies are intuitively meaningful. Annotator-level variability is also notable: the weakest alignment consistently comes from the annotator whose academic background differed most from the rest of the group, which may be partially explained by the domain composition of the evaluation set, where five of the fifteen target concepts were drawn from the computer science domain.

Table 3 compares Claude’s implicit ranking of analogies with human rankings. In contrast to the dimension-level results, ranking agreement is generally stronger, with most annotators showing moderate-to-high correlation (ρ between 0.46 and

Annotator	ρ	p-value	Significant
ANN-6JJP46	0.474	0.0010	True
ANN-6LG7WY	0.705	0.0000	True
ANN-DUNVTH	0.307	0.0404	True
ANN-H2NTMS	0.694	0.0000	True
ANN-KKN85E	0.463	0.0014	True
ANN-Q777H6	0.128	0.4012	False
ANN-VM9J6N	0.694	0.0000	True

Table 3: Spearman correlation between Claude’s implicit ranking and human rankings.

0.70) and statistically significant results ($p < 0.05$). The strongest alignment is observed for ANN-6LG7WY ($\rho = 0.705$), while ANN-Q777H6 again shows no significant agreement. These results indicate that Claude more reliably captures relative preferences between analogies than assigning consistent absolute scores, suggesting that LLM-as-a-judge is better suited for comparative evaluation settings than for fine-grained scoring.

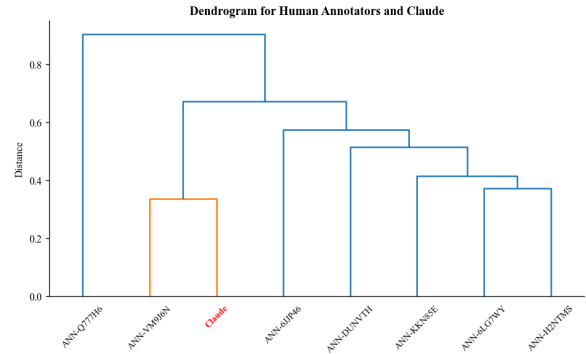


Figure 5: Hierarchical clustering of human annotators and Claude based on their scoring patterns across evaluation dimensions. Distances reflect dissimilarity in judgments (computed from Spearman correlations). Claude clusters closely with a subset of annotators (e.g., ANN-VM9J6N), while other annotators (e.g., ANN-Q777H6) form more distant branches.

Figure 5 further examines agreement patterns through hierarchical clustering of annotators and Claude. The dendrogram reveals that Claude groups closely with a subset of annotators, suggesting similar evaluation behavior, while others form more distant clusters, indicating systematic differences in scoring. In particular, annotators with lower correlation scores (e.g., ANN-Q777H6) appear as outliers, consistent with the results in Table 3. This analysis reinforces that, although Claude aligns well with the majority of annotators, human judgments themselves are not fully consistent, highlighting the inherent subjectivity of analogy evaluation.

5 Conclusion

We presented a modular pipeline for educational analogy generation decomposed into source finding, sub-concept generation, explanation generation, and evaluation, systematically evaluated across 12 LLMs and 7 embedding models on SCAR and ParallelPARC. A central finding is that the best model and configuration differ across stages: Grok-4-Fast leads sub-concept matching without background descriptions, while LLaMA-3.1-70B with background is best for generation; GPT-4.1-Mini with unpaired sub-concept mappings achieves the strongest explanation quality; and OpenAI large embeddings with sub-concepts lead closed-setting retrieval, whereas the target-only configuration is more practical in the open setting. Sub-concept grounding consistently improves retrieval precision and explanation quality but provides limited benefit in open-ended source generation. For evaluation, DeepSeek-R1 and Grok-4-Fast produce the highest-quality analogies, and Claude Sonnet 4.6 aligns more reliably with human rankings than absolute scores, suggesting LLM-as-a-judge is better suited to comparative settings. Taken together, these findings demonstrate that no single model dominates across all stages, reinforcing the necessity of a modular, stage-aware approach to analogy generation.

Limitations

Several limitations of this work warrant acknowledgment. First, all experiments are conducted exclusively on English-language data, leaving the pipeline’s behaviour on morphologically richer or structurally different languages unexplored. In such settings, both embedding-based retrieval and generation quality may differ, and evaluation metrics such as Semantic Match Accuracy (SMA) may require recalibration.

Second, the 0.7 cosine similarity threshold used for SMA is a fixed heuristic selected for MiniLM-L6 embeddings rather than empirically validated. Its suitability for more abstract or diverse sub-concepts remains unclear, and alternative thresholds could meaningfully affect generation results.

Third, while each pipeline stage is evaluated independently, we do not evaluate the pipeline end-to-end. In a full pipeline setting, errors are likely to accumulate and propagate across stages. For example, an incorrect source selected during the retrieval stage would lead to flawed sub-concept mappings

and explanations downstream. This makes source finding a particularly critical stage, as early errors can disproportionately affect final analogy quality. Future work should investigate strategies to mitigate such compounding effects, such as iterative refinement, confidence-aware filtering, or alternative candidate selection mechanisms.

Fourth, the current architecture relies primarily on dense embedding models for retrieval and does not explore hybrid search strategies. Combining dense and sparse retrieval methods, or incorporating re-ranking models more extensively, could improve the robustness and quality of source selection, particularly in open-ended settings.

Fifth, analogy quality is inherently subjective: an analogy that is effective for one learner may not be equally meaningful for another. While this subjectivity is partially reflected in moderate inter-annotator agreement, our evaluation does not explicitly model user-dependent preferences. This limits the ability to assess how well generated analogies generalise across diverse learner populations. Future work should investigate how to construct learner profiles and incorporate them into the pipeline to generate personalised analogies tailored to factors such as cultural background, age, prior knowledge, and individual interests or hobbies.

Sixth, all pipeline stages are evaluated in a single-pass setting. We do not measure output variance across runs, which means that reported results reflect point estimates rather than stable averages. Although a low temperature (0.2) is used, some variability is expected, particularly for smaller models. Moreover, since ParallelPARC was used in only one stage of the pipeline (embedding-based retrieval) and the remaining stages relied solely on SCAR, the generalizability of the results across datasets may be limited.

Finally, the human evaluation study involves six out of seven graduate-level annotators from a single institution, which may introduce demographic and disciplinary homogeneity. Inter-annotator agreement is moderate, reflecting the inherent subjectivity of analogy evaluation and suggesting that a larger, more diverse annotator pool could yield different conclusions. In addition, open-setting source finding is evaluated only on SCAR due to computational constraints, limiting the generalisability of these findings to other domains of different difficulty.

Ethical Considerations

We do not anticipate any ethical risks related to this study.

References

- Briana Bettin and Linda Ott. 2023. Pedagogical prisms: toward domain isomorphic analogy design for relevance and engagement in computing education. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, pages 410–416.
- Bhavya Bhavya, Chris Palaguachi, Yang Zhou, Suma Bhat, and ChengXiang Zhai. 2024. Long-form analogy evaluation challenge. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 1–16.
- David E Brown and John Clement. 1989. Overcoming misconceptions via analogical reasoning: Abstract transfer versus explanatory model construction. *Instructional science*, 18(4):237–261.
- Tamara Czinczoll, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2022. Scientific and creative analogies in pretrained language models. *arXiv preprint arXiv:2211.15268*.
- Stina Eriksson, Niklas Gericke, and Karin Thörne. 2024. Analogy competence for science teachers. *Studies in Science Education*, pages 1–29.
- Dedre Gentner. 1981. Are scientific analogies metaphors? Technical report.
- Cheng Jiayang, Lin Qiu, Tsz Ho Chan, Tianqing Fang, Weiqi Wang, Chunkit Chan, Dongyu Ru, Qipeng Guo, Hongming Zhang, Yangqiu Song, and 1 others. 2023. Storyanalogy: Deriving story-level analogies from large language models to unlock analogical understanding. *arXiv preprint arXiv:2310.12874*.
- Maurice G Kendall. 1990. *Rank Correlation Methods*, 5th edition. Oxford University Press.
- Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.
- 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. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.
- Jeongyeon Kim, Sangho Suh, Lydia B Chilton, and Haijun Xia. 2023. Metaphorian: Leveraging large language models to support extended metaphor creation for science writing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, pages 115–135.
- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability. *Annenberg School for Communication Departmental Papers*.
- Nianqi Li, Siyu Yuan, Jiangjie Chen, Jiaqing Liang, Feng Wei, Zujie Liang, Deqing Yang, and Yanghua Xiao. 2024. Past meets present: Creating historical analogy with large language models. *arXiv preprint arXiv:2409.14820*.
- 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.
- Zekai Shao, Siyu Yuan, Lin Gao, Yixuan He, Deqing Yang, and Siming Chen. 2025. Unlocking scientific concepts: How effective are llm-generated analogies for student understanding and classroom practice? In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19.
- Oren Sultan, Yonatan Bitton, Ron Yosef, and Dafna Shahaf. 2024. Parallelparc: A scalable pipeline for generating natural-language analogies. *arXiv preprint arXiv:2403.01139*.
- Niket Tandon, Bhavana Dalvi Mishra, Joel Grus, Wentau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. *arXiv preprint arXiv:1808.10012*.
- Xiao Ye, Andrew Wang, Jacob Choi, Yining Lu, Shreya Sharma, Lingfeng Shen, Vijay Tiyyala, Nicholas Andrews, and Daniel Khashabi. 2024. Analobench: benchmarking the identification of abstract and long-context analogies. *arXiv preprint arXiv:2402.12370*.
- Siyu Yuan, Jiangjie Chen, Xuyang Ge, Yanghua Xiao, and Deqing Yang. 2023a. Beneath surface similarity: Large language models make reasonable scientific analogies after structure abduction. *arXiv preprint arXiv:2305.12660*.
- Siyu Yuan, Jiangjie Chen, Changzhi Sun, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. 2023b. Analogykb: Unlocking analogical reasoning of language models with a million-scale knowledge base. *arXiv preprint arXiv:2305.05994*.
- ChengXiang Zhai, Suma Bhat, and Heng Ji. 2024. Long-form analogy extraction, generation, and evaluation with large language models.
- Xiaofei Zhou, Yunfan Gong, Yushan Zhou, Yufei Jiang, and Zhen Bai. 2025. Co-design of analogical and embodied representations with children for child-centered ai learning experiences. *International Journal of Human-Computer Studies*, 199:103462.

A Detailed Record of Datasets

Listing 1 and 2 shows a representative JSON example of the two datasets.

```
{
  "id": 3,
  "lang": "en",
  "system_a": "Respiratory system",
  "system_b": "Engine",
  "mappings": [
    ["oxygen", "fuel"],
    ["lungs", "combustion chamber"],
    ["breathing muscles", "piston"]
  ],
  "system_a_domain": "Biology",
  "system_b_domain": "Physics",
  "system_a_background": "The respiratory system ...",
  "system_b_background": "An engine or motor ...",
  "Explanation": [
    "Oxygen corresponds to fuel: ...",
    "Lungs correspond to the combustion chamber: ...",
    "Breathing muscles correspond to the piston: ..."
  ]
}
```

Listing 1: Representative JSON example from SCAR

```
{
  "sample_id": 1,
  "source_subject": "What causes a volcano to erupt?",
  "source_domain": "Natural Sciences",
  "target_domain": "Engineering",
  "target_subject": "What causes a boiler to explode?",
  "target_field": "Chemical Engineering",
  "relations": [
    "(magma, heats, underground water) like (steam, heats, liquid)",
    "(pressure, builds, inside the volcano) like (pressure, builds, inside the boiler)",
    "(magma, pushes, against the walls of the volcano) like (steam, pushes, against the walls of the boiler)"
  ],
  "source_paragraph": "When magma heats up underground water, pressure begins to build up inside the volcano, leading to an eruption.",
  "target_paragraph": "Steam heats the liquid inside the boiler, causing the pressure to build up until the boiler can no longer contain it.",
  "analogy_type": "far analogy",
  "sum_vote_analogy": 3.0
}
```

Listing 2: Representative JSON from ParallelPARC

B Prompt for LLM Re-ranker

You are selecting the best analogous source concepts for a scientific analogy.

Your task:

1. Analyze the target concept and its properties
2. Review each candidate source and its generated analogous properties
3. Select the 3 candidates whose properties BEST map to the target properties

Selection criteria:

- Strong structural/functional correspondence between source and target properties
- The source concept should be familiar and help explain the unfamiliar target
- Prefer sources with clear, well-mapped properties over vague ones

Return the EXACT names of your top 3 selected candidates.

""

Listing 3: Prompt used to select the top-3 analogies from the top-10 potential sources. This was wrapped in DSPY function as instruction, alongside specific input and output variables.

C Threshold Decision for Semantic Hit@K

To define a threshold for semantic Hit@K, we analyse the distribution of source-to-source similarity scores in the SCAR dataset. Specifically, we compare similarity values between *correct* source pairs (i.e., gold analogies) and *incorrect* pairs obtained by randomly shuffling sources.

Figure 6 shows that correct source pairs tend to exhibit higher similarity than incorrect ones, but with substantial overlap between the two distributions. This overlap indicates that a fixed high threshold (e.g., 0.5) would be overly restrictive, excluding many valid analogies that are structurally sound but not highly similar at the surface level.

To balance precision and coverage, we adopt a data-driven threshold of 0.374, corresponding to the upper tertile (top one-third) of similarity scores among gold source pairs. This choice captures a substantial portion of valid analogies while filtering out a large fraction of incorrect ones. As shown in the figure, this threshold lies above the mean similarity of incorrect pairs (0.180) and close to the mean of correct pairs (0.368), providing a principled trade-off between false positives and false negatives.

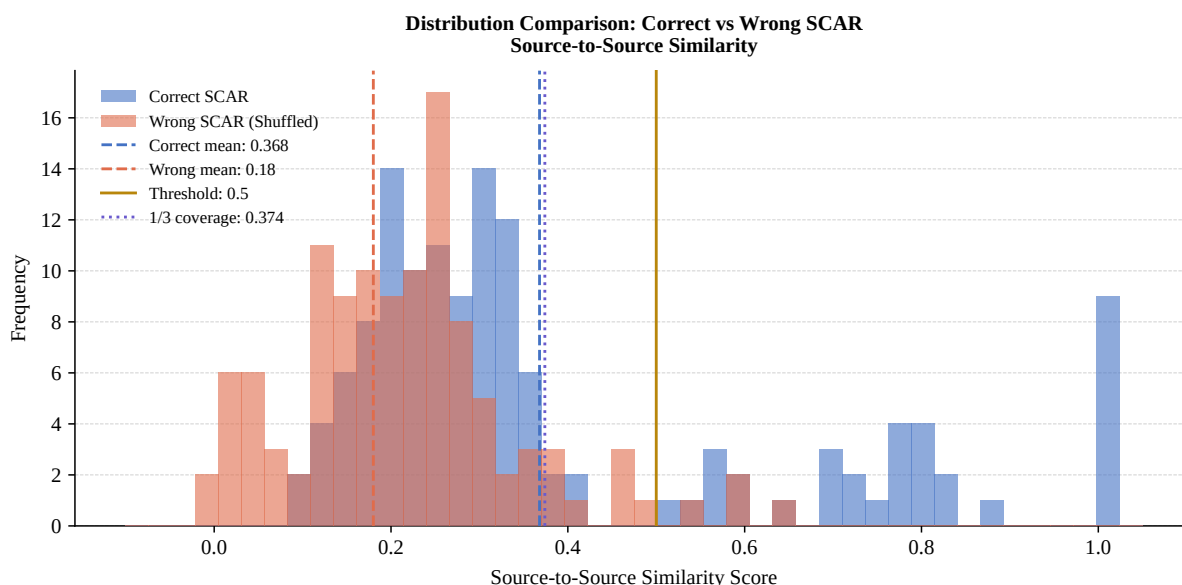


Figure 6: Distribution of source-to-source similarity scores in SCAR for correct (gold) and incorrect (shuffled) pairs. Dashed lines indicate the mean similarity for each group, while the dotted line shows the chosen threshold (0.374), corresponding to the upper tertile of gold similarities. A higher fixed threshold (e.g., 0.5) would exclude many valid analogies, motivating the use of a data-driven cutoff.

D Explanation Generation Prompts

This appendix lists the DSPy signatures used for the six explanation-generation settings. Each signature corresponds to one input configuration.

Setting 1: System Names Only This configuration provides only the target and source system names.

```
class ExplGen_Base(dspy.Signature):
    """Generate an analogy explanation using system names only."""
    target: str = dspy.InputField(
        desc="Target system"
    )
    source: str = dspy.InputField(
        desc="Source system"
    )
```

```

)
explanation: str = dspy.OutputField(
    desc="Explanation of the analogy"
)

```

Setting 2: System Names with Background Descriptions This setting augments system names with background descriptions for both systems.

```

class ExplGen_BaseDesc(dspy.Signature):
    """Generate an analogy explanation using names and descriptions."""
    target: str = dspy.InputField(
        desc="Target system"
    )
    target_desc: str = dspy.InputField(
        desc="Background description of the target"
    )
    source: str = dspy.InputField(
        desc="Source system"
    )
    source_desc: str = dspy.InputField(
        desc="Background description of the source"
    )
    explanation: str = dspy.OutputField(
        desc="Explanation of the analogy"
    )

```

Setting 3: System Names with Unpaired Sub-concepts This configuration provides both systems with unpaired lists of sub-concepts.

```

class ExplGen_Sub(dspy.Signature):
    """Generate an analogy explanation using unpaired sub-concepts."""
    target: str = dspy.InputField(
        desc="Target system"
    )
    target_sub: list[str] = dspy.InputField(
        desc="Target sub-concepts"
    )
    source: str = dspy.InputField(
        desc="Source system"
    )
    source_sub: list[str] = dspy.InputField(
        desc="Source sub-concepts"
    )
    explanation: list[str] = dspy.OutputField(
        desc="Explanation after implicitly pairing sub-concepts"
    )

```

Setting 4: System Names with Unpaired Sub-concepts and Descriptions This setting extends Setting 3 by adding background descriptions for both systems.

```

class ExplGen_SubDesc(dspy.Signature):
    """Generate an analogy explanation using unpaired sub-concepts and descriptions."""
    target: str = dspy.InputField(
        desc="Target system"
    )
    target_desc: str = dspy.InputField(
        desc="Background description of the target"
    )
    target_sub: list[str] = dspy.InputField(
        desc="Target sub-concepts"
    )
    source: str = dspy.InputField(
        desc="Source system"
    )
    source_desc: str = dspy.InputField(
        desc="Background description of the source"
    )

```

```

source_sub: list[str] = dspy.InputField(
    desc="Source sub-concepts"
)
explanation: list[str] = dspy.OutputField(
    desc="Explanation after implicitly pairing sub-concepts"
)

```

Setting 5: Paired Sub-concept Mappings This configuration provides explicit one-to-one sub-concept alignments between the target and source systems.

```

class ExplGen_Pair(dspy.Signature):
    """Generate an analogy explanation using paired sub-concepts."""
    target: str = dspy.InputField(
        desc="Target system"
    )
    source: str = dspy.InputField(
        desc="Source system"
    )
    pairs: list[list[str]] = dspy.InputField(
        desc="Paired target-source sub-concepts"
    )
    explanation: list[str] = dspy.OutputField(
        desc="Explanation for each paired sub-concept"
    )

```

Setting 6: Paired Sub-concept Mappings with Descriptions This setting combines explicit sub-concept pairings with background descriptions for both systems.

```

class ExplGen_PairDesc(dspy.Signature):
    """Generate an analogy explanation using paired sub-concepts and descriptions."""
    target: str = dspy.InputField(
        desc="Target system"
    )
    target_desc: str = dspy.InputField(
        desc="Background description of the target"
    )
    source: str = dspy.InputField(
        desc="Source system"
    )
    source_desc: str = dspy.InputField(
        desc="Background description of the source"
    )
    pairs: list[list[str]] = dspy.InputField(
        desc="Paired target-source sub-concepts"
    )
    explanation: list[str] = dspy.OutputField(
        desc="Explanation for each paired sub-concept"
    )

```

E Prompt for the LLM-as-a-Judge

You are an expert evaluator of scientific analogies.

Given a target concept and a chosen source analogy, evaluate whether this is a good analogy. A good analogy uses a FAMILIAR source concept to explain an UNFAMILIAR target concept through meaningful structural or functional parallels.

For each of the three dimensions below, first provide a brief reasoning explaining your assessment, then give the numeric score (1, 2, or 3).

ANALOGY_COHERENCE: Does the pairing make intuitive sense?

- 3: The connection is immediately clear and natural -- most people would see it without explanation. The source and target share an obvious structural or functional parallel.
- 2: A meaningful connection exists but requires some explanation to see. The link is real but not self-evident; a sentence or two is needed to establish it.
- 1: No meaningful connection exists, or the pairing is random, forced, or misleading.

MAPPING_SOUNDNESS: Could properties/mechanisms of the source map to the target?

- 3: Rich, consistent structural or functional correspondences exist.
Multiple source properties map precisely onto target properties.
The mapping holds across the main components of both domains.
- 2: Some valid mappings exist, but coverage is partial.
Core correspondences work, but important aspects of the target are not represented, or some mappings are approximate or strained.
- 1: No valid mappings are possible, or the apparent mappings are entirely superficial or misleading. Source and target are fundamentally incompatible.

EXPLANATORY_POWER: Would this analogy help a learner understand the target?

- 3: The analogy clearly illuminates the target concept and supports correct reasoning.
A learner could use it to predict or explain target behavior.
- 2: The analogy provides partial insight with notable limitations.
It conveys the general idea but cannot support deeper reasoning, or it risks creating minor misconceptions.
- 1: The analogy fails to aid understanding and would likely confuse or mislead a learner.

CALIBRATION EXAMPLES -- use these to anchor your scoring.

Score each dimension independently. The scale is 1-3 only:

3 = strong / clearly works

2 = partial / works with caveats

1 = doesn't work / poor or misleading

Example 1 (scores: 3 / 3 / 3)

target: "electric circuit" | analogy: "water flowing through pipes"

coherence=3:

The connection is immediately clear without any explanation needed.

A driving force pushes a substance through a constrained pathway in both cases -- this structural parallel is instantly visible.

mapping=3:

Multiple precise correspondences hold consistently:

voltage -> pressure | current -> flow rate | resistance -> pipe restriction

battery -> pump | closed loop -> closed pipe system

Relationships between variables (increase pressure -> increase flow) are preserved.

explanatory=3:

A learner can use the analogy to reason about Ohm's law qualitatively and make correct predictions about circuit behavior.

Example 2 (scores: 3 / 2 / 3)

target: "cell" | analogy: "factory"

coherence=3:

The analogy captures organized division of labor within a bounded system.

Most people immediately see why a cell resembles a factory.

mapping=2:

nucleus -> control center | ribosomes -> production workers

mitochondria -> energy supply | membrane -> boundary/security

explanatory=3:

Helps learners grasp coordination and specialization clearly.

Example 3 (scores: 2 / 2 / 2)

target: "mathematical function" | analogy: "machine"

coherence=2:

The shared idea (input -> transformation -> output) is real but requires explanation.

mapping=2:

input -> raw material, rule -> mechanism, output -> product

explanatory=2:

Useful for initial intuition but limited for deeper reasoning.

Example 4 (scores: 1 / 1 / 1)

target: "neural network" | analogy: "human brain"

coherence=1:
Superficial similarity ("neurons") without structural alignment.

mapping=1:
No valid correspondences across core mechanisms.

explanatory=1:
Likely to mislead understanding.

Example 5 (scores: 1 / 1 / 1)
target: "chemical reaction" | analogy: "a novel"

coherence=1:
No meaningful connection.

mapping=1:
No systematic correspondences.

explanatory=1:
No instructional value.

Example 6 (scores: 3 / 1 / 2)
target: "democracy" | analogy: "majority vote in a classroom"

coherence=3:
Both involve majority-based decisions.

mapping=1:
Institutional aspects have no counterpart.

explanatory=2:
Partial understanding only.

Example 7 (scores: 2 / 3 / 3)
target: "atom" | analogy: "solar system"

coherence=2:
Spatial similarity exists but requires explanation.

mapping=3:
nucleus -> sun | electrons -> planets

explanatory=3:
Supports initial reasoning about structure.

Example 8 (scores: 3 / 2 / 1)
target: "photosynthesis" | analogy: "solar-powered factory"

coherence=3:
Energy conversion is clearly analogous.

mapping=2:
sunlight -> energy supply | chloroplast -> factory

explanatory=1:
Oversimplifies processes.

Example 9 (scores: 1 / 3 / 3)
target: "compound interest" | analogy: "a snowball rolling downhill"

coherence=1:
Domains differ fundamentally.

mapping=3:
principal -> snowball | growth -> accumulation

explanatory=3:
Clearly conveys accelerating growth.

```
Example 10 (scores: 3 / 1 / 2)
target: "ecosystem" | analogy: "a family"

coherence=3:
  Interdependence is clear.

mapping=1:
  No structural ecological correspondences.

explanatory=2:
  Partial intuition only.
```

Listing 4: Prompt used to evaluate the analogies from different settings on different dimensions. This was wrapped in DSPY function as instruction, alongside specific input and output variables.

F Ablation results for Sub-concept Matching

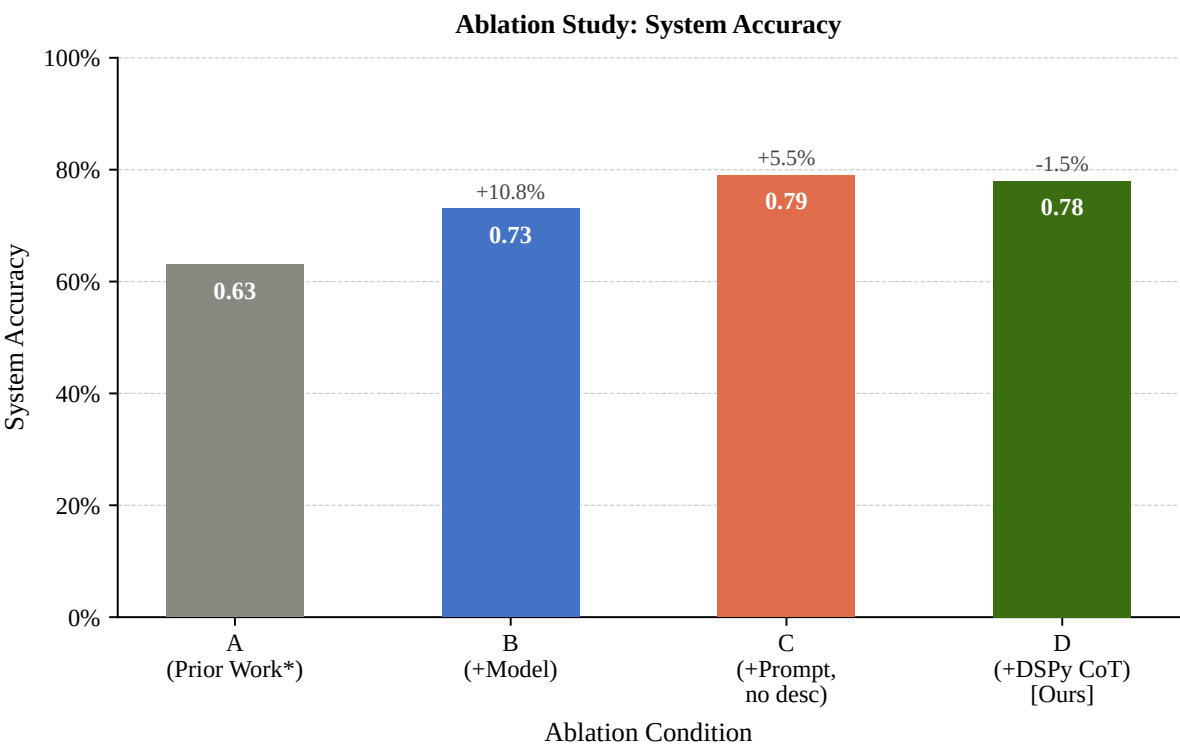


Figure 7: Ablation study on sub-concept matching system accuracy, isolating the contribution of each component relative to prior work (Yuan et al., 2023a). Upgrading the model to our best performing Grok-4 alone (B) yields the largest gain (+10.8%), followed by prompt redesign (C, +5.5%). Wrapping with DSPy Chain-of-Thought (D, our final system) introduces a marginal drop (−1.5%), confirming that automated prompt optimisation does not improve over the hand-crafted prompt for this task.

Figure 7 shows the ablation study to figure out what aspect actually led to the improved results.

G Explanation Heatmap Results

Figure 8 presents SBERT similarity scores across twelve LLMs and six experimental settings for the explanation generation stage.

H Open Setting Source Finding: Per-Model Hit@20 Comparison

Figure 9 presents per-model Hit@20 scores in the open setting, comparing target-only and target with sub-concept prompting across all twelve models.

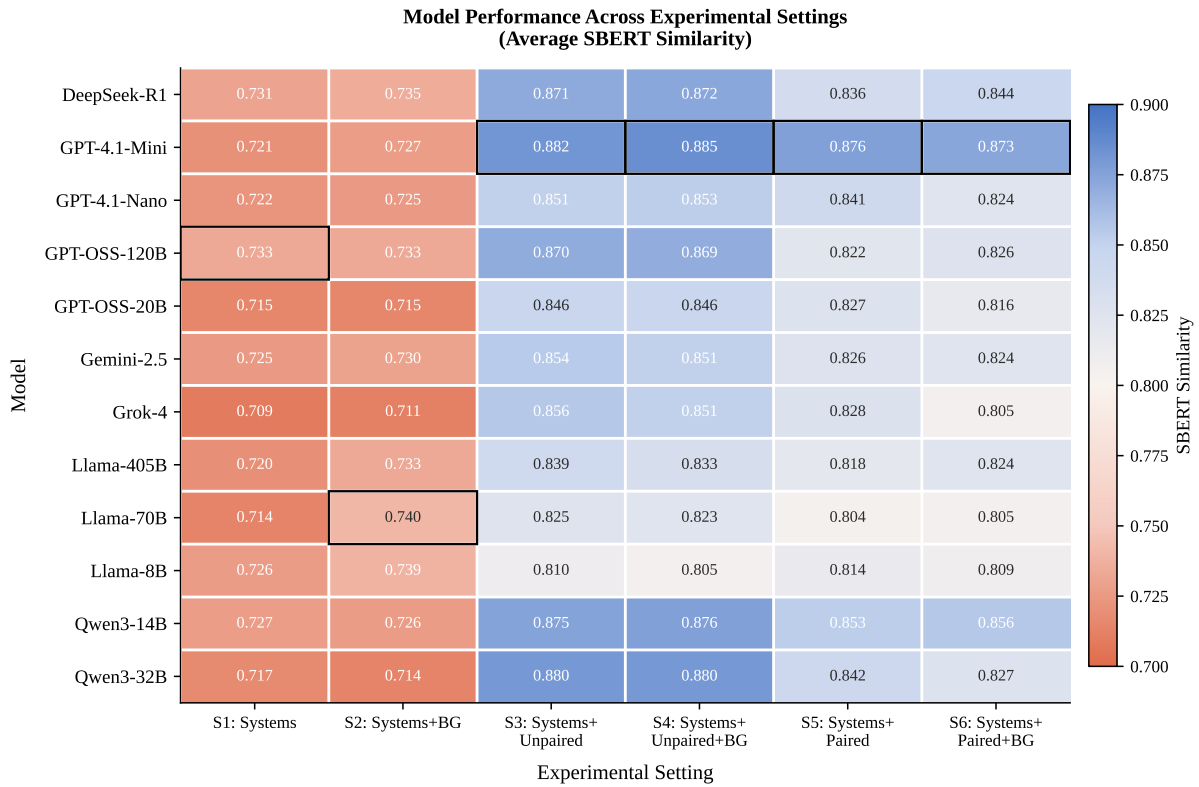


Figure 8: Average SBERT similarity scores across twelve models and six experimental settings. Settings S1-S2 provide only system-level context (without and with background descriptions, respectively), while S3-S4 additionally supply unpaired sub-concepts and S5-S6 supply paired sub-concepts. Black borders indicate the best-performing model per setting. GPT-4.1-Mini consistently leads in sub-concept-enriched settings (S3-S5), while DeepSeek-R1 and Qwen3 variants are competitive in paired settings (S5-S6). Scores rise substantially when sub-concepts are introduced (S3-S6 vs. S1-S2), suggesting that grounding the comparison in explicit sub-concept structure is a stronger driver of similarity than background context alone.

I Embedding-based Retrieval Results

Figure 10 and 11 show the heatmap of the 7 embedding models across 4 input configurations.

J Annotation Dataset

Table 4 presents the annotation dataset, comprising 15 target concepts spanning five domains, each paired with three source analogies evaluated under target-only and target with sub-concept conditions. Furthermore, Figures 12- 21 provide an overview of the web-based annotation form used in our human evaluation study. The form guides annotators through rating each of three candidate source analogies per target concept on three dimensions – coherence, mapping soundness, and explanatory power – on a 1–3 scale, before ranking the three sources by overall learning usefulness and reporting their confidence in the ranking.

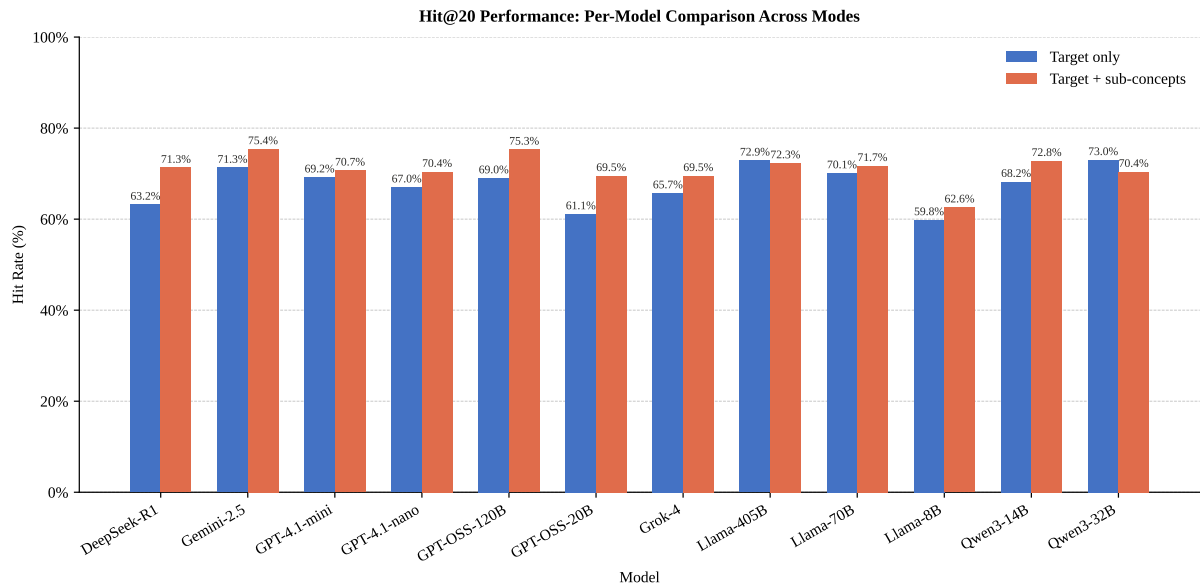


Figure 9: Per-model Hit@20 comparison in the open setting under Target-Only and Target + Sub-Concept modes. Results are reported using semantic matching and highlight the effect of incorporating sub-concepts on top-20 retrieval performance.

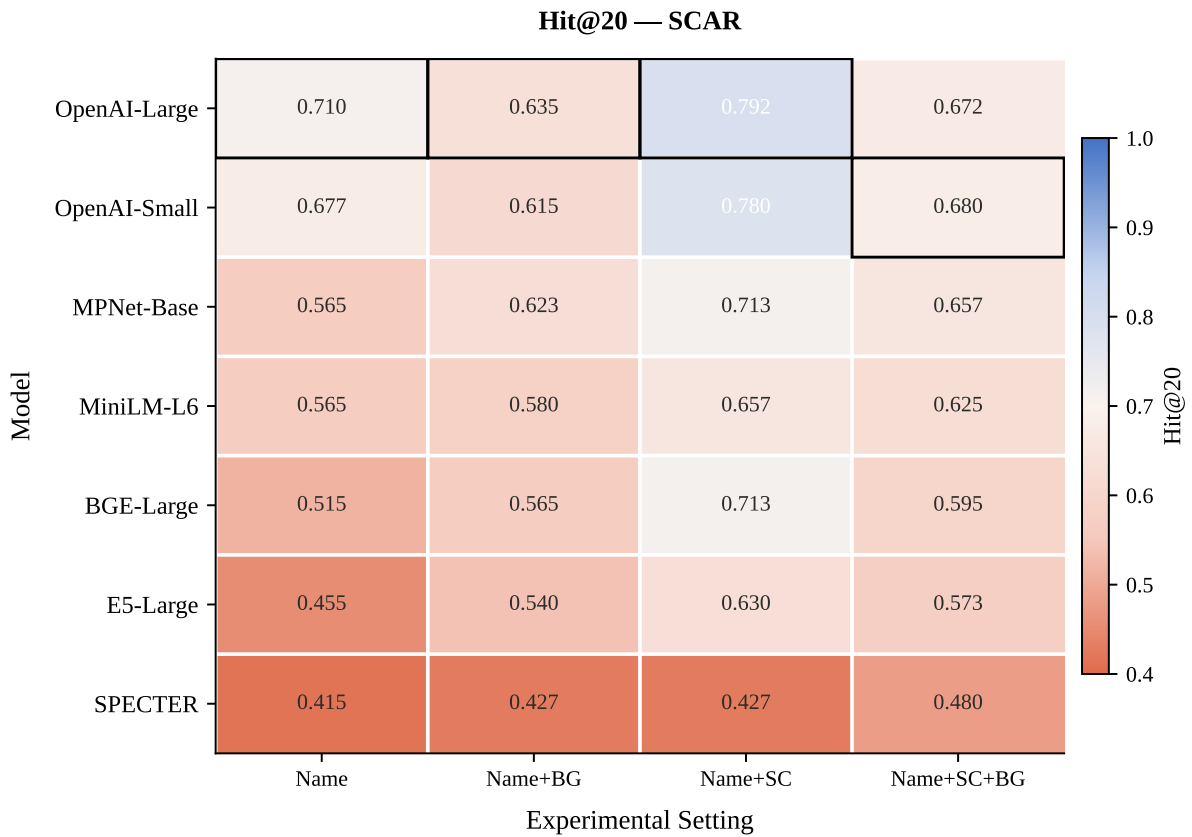


Figure 10: Hit@20 performance on the SCAR dataset across embedding models and input configurations. Incorporating sub-concepts consistently improves retrieval compared to name-only inputs, while adding background provides mixed benefits. OpenAI embeddings achieve the strongest overall performance.

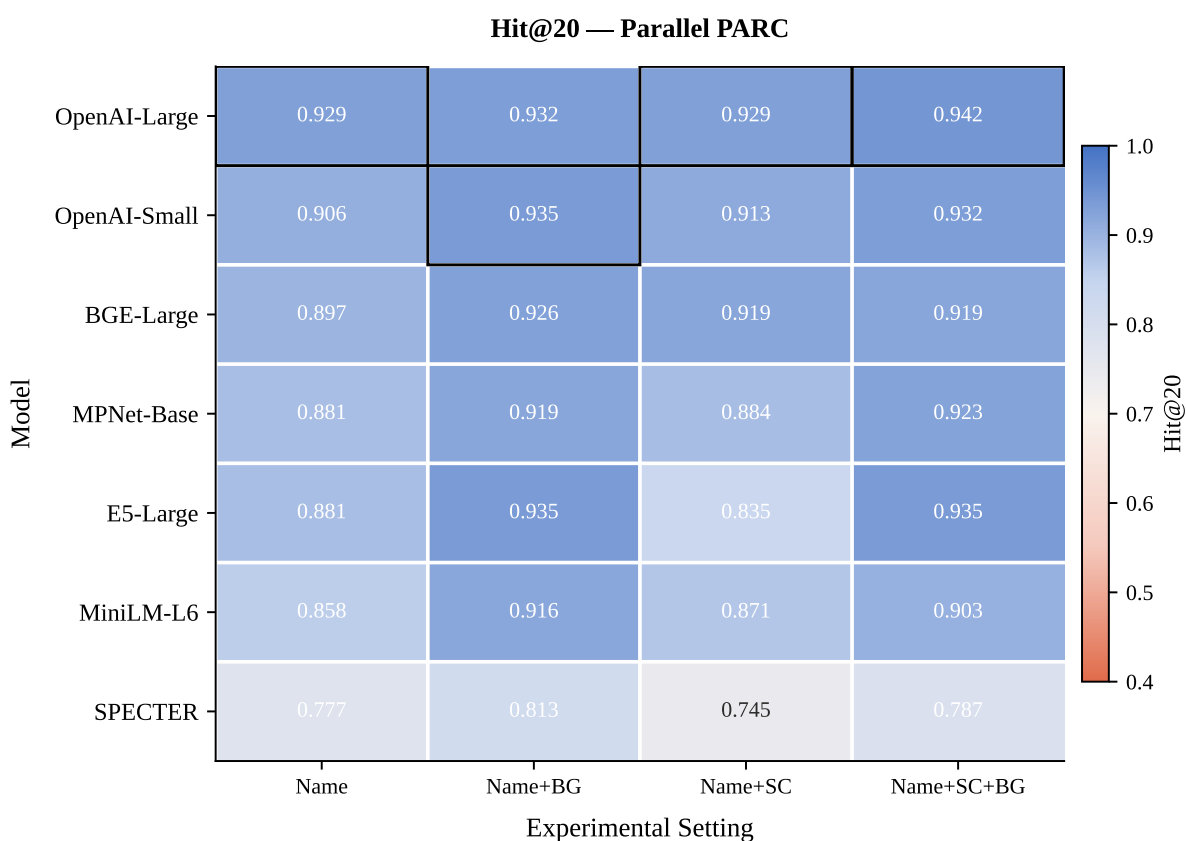


Figure 11: Hit@20 performance on the ParallelPARC dataset. Retrieval performance is generally higher than SCAR, reflecting the more structured nature of process-based analogies. Combining sub-concepts with background yields the best results across most models, indicating the value of richer contextual representations.

Table 4: Annotation dataset: 15 target concepts across five domains, each paired with three source analogies. T = Target-only, T+S = Target + Sub-concepts.

#	Target	Domain	Source Analogy	Model	Mode
1	Gas Diffusion	Chemistry	Perfume spreading Smoke dispersion Gasoline	GPT-OSS-120B Qwen3-14B Qwen3-14B	T T+S T
2	Encryption	CS	locked box Sealed envelope Locker	GPT-OSS-120B Grok-4-fast GPT-4.1-nano	T T+S T+S
3	Industrial Revolution	History	Steam engine era Automation Industrialization	GPT-OSS-120B LLaMA-8B LLaMA-405B	T T+S T
4	GAN Algorithm	CS	Art forgery Artist and Critic Teacher	Grok-4-fast Grok-4-fast GPT-4.1-nano	T T+S T
5	Bill of Rights	Law	Rulebook New-car warranty School curfew	Qwen3-32B GPT-OSS-20B GPT-OSS-20B	T T T
6	AdaBoost Algorithm	CS	Teamwork iterative training Gears in a machine	Gemini-2.5 GPT-OSS-120B LLaMA-70B	T+S T T+S
7	Platelets	Biology	Tiny repair crew Band-Aid Beads	Gemini-2.5 Grok-4-fast GPT-4.1-mini	T T+S T
8	Cell	Biology	factory House Garage	Grok-4-fast LLaMA-405B LLaMA-8B	T T T+S
9	Economics	Economics	Marketplace Household Budget Ecosystem	Qwen3-14B LLaMA-70B Qwen3-14B	T T T+S
10	Chemical Reaction	Chemistry	Baking soda + vinegar Digestion Boiling water	Qwen3-14B Qwen3-32B GPT-4.1-mini	T T+S T
11	Photosynthesis	Biology	Solar panels Greenhouse farming Fermentation Process	LLaMA-405B Grok-4-fast LLaMA-70B	T+S T T+S
12	Random forest	CS	Voting committee Jury deliberation Tree	Gemini-2.5 GPT-OSS-120B GPT-4.1-nano	T+S T+S T
13	Isotope Dating	Chemistry	Hourglass Atomic Clock Half-Life	DeepSeek-R1 LLaMA-405B LLaMA-70B	T+S T T
14	Renaissance	History	spring bloom Ancient Rome Enlightenment	GPT-OSS-120B retrieval-based LLaMA-70B	T Embedding T+S
15	EM Algorithm	CS	Detective investigation guess-and-check Miner	GPT-OSS-120B GPT-OSS-120B retrieval-based	T T+S Embedding

Welcome to the Analogy Quality Annotation Study

Your expert judgment helps us validate how well automated systems evaluate analogies for education.

⚠ Important: Please complete the form in one sitting, as refreshing the page will reset your progress.

BACKGROUND

We are building a system that generates **analogies** to help learners understand complex concepts through familiar comparisons. Large Language Models (LLMs) serve as automated judges to rate analogy quality — but we need **human evaluation** to verify their reliability.

YOUR TASK

You will see **15 target concepts**, each paired with **3 candidate source analogies**.

All 15 Targets

Rate each analogy on 3 quality dimensions (1–3 scale), rank the 3 analogies by learning usefulness, and indicate your confidence in the ranking.

Estimated time: **60–90 minutes**. A floating reference guide will be available throughout.

← Back

Continue →

Figure 12: (a) Welcome screen introducing the task and study background.

Key Definitions & Rating Scale

Read these carefully — they define exactly what you're evaluating.

WHAT IS AN ANALOGY?

An **analogy** maps a *target concept* (the unfamiliar topic being learned) onto a *source concept* (something more familiar). A good analogy preserves structural and functional relationships, helping learners transfer understanding from the known to the unknown.

Example: "A **cell** is like a **factory**" — Cell is the target, Factory is the source.

THE THREE DIMENSIONS

1. COHERENCE

Does the pairing make intuitive sense? When you hear "Target is like Source," does the connection feel natural and immediately understandable?

2. MAPPING

Do the key parts of the target correspond to parts of the source? Read both descriptions carefully and try to identify counterparts between important components, processes, or relationships of the target and the source.

How to identify mappings: Go through the target description and list its key elements (components, roles, processes, relationships, inputs, outputs). Then, for each element, ask: *is there something in the source description that plays a similar role or serves a similar function?* The more elements you can pair up, the higher the mapping score. Elements left without a counterpart count against it.

Tip — look for shared roles (e.g., something that acts as a barrier in both), *shared processes* (e.g., both involve something flowing or transforming), and *shared relationships* (e.g., one part controlling another in both).

Note on cross-type analogies: Some analogies compare a process to an object or device (e.g., comparing a biological process to a machine). This is perfectly fine — what matters is whether the **functional roles and**

Figure 13: (b) Key definitions and rating scale for the three evaluation dimensions.

Note on cross-type analogies: Some analogies compare a process to an object or device (e.g., comparing a biological process to a machine). This is perfectly fine — what matters is whether the **functional roles and relationships** align, not whether both are the same kind of entity. A device can have parts that map onto the stages or components of a process and vice versa.

3. EXPLANATORY POWER

Would this analogy actually help someone learn the target concept? Does it make the abstract concrete, reveal causal relationships, or enable the learner to reason about how the target works?

A key factor is the **complexity gap** between source and target. The most useful analogies explain complex or unfamiliar concepts through simpler, more concrete, more everyday sources — moving the learner from the known to the unknown. An analogy where the source is just as complex or technical as the target may have good structural correspondence, but limited teaching value because the learner has no easier foothold to start from. For example, explaining one scientific process through another equally specialized process offers less educational benefit than explaining it through something a non-expert already understands intuitively.

When scoring Explanatory Power, consider: is the source meaningfully simpler or more familiar than the target? A "High" score typically requires the source to give the learner an easier entry point — not just a structurally parallel one.

RATING SCALE

SCORE	LEVEL	GUIDELINE
3	High	Strong and effective. Little to no weakness on this dimension.
2	Medium	Partial or acceptable. Some elements work but with notable gaps.
1	Low	Weak or broken. The analogy fails on this dimension.

Figure 14: (c) Explanatory power dimension definition and full rating scale.

RANKING CONFIDENCE

After ranking the three source analogies for each target, you will indicate your **confidence in the ranking**. For some triplets, the quality differences are stark and the order is obvious; for others, the analogies are very close in quality and the ranking feels like a coin flip.

LEVEL	MEANING
High	The quality differences are clear — this ranking order is obvious.
Medium	You have a reasonable sense of the order, but some positions were close calls.
Low	The analogies are very similar in quality — the ranking could easily go another way.

WORKED EXAMPLES (NOT PART OF YOUR ANNOTATION)

Three examples showing how to rate analogies. Click each tab to study a different target. Note: in these examples we describe the reasoning behind each score — during your annotation, you should apply similar reasoning by reading the source and target descriptions and identifying parallels yourself.

Atom

DNA

Electricity

Target: Atom (Physics) — The fundamental unit of matter, consisting of a dense central nucleus (protons and neutrons) surrounded by electrons in orbital shells, held together by electromagnetic forces.

"Solar System" → Coh 3 · Map 3 · Exp 3

Coh 3: The image of a central mass with orbiting bodies clicks immediately with the idea of a nucleus surrounded by electrons.

Map 3: Nearly every key component of the atom has a natural counterpart in the solar system — the central body, the orbiting objects, the governing force, and the orbital structure all find clear parallels.

Exp 3: A learner can use this analogy to reason about spatial layout, the role of forces, and relative scale.

"Marble in a Stadium" → Coh 2 · Map 2 · Exp 2

Coh 2: Conveys emptiness and scale contrast, but the full atomic structure isn't immediately apparent.

Map 2: The small object inside a vast space maps onto one aspect (the nucleus in mostly empty space), but several key parts of the atom — the orbiting particles, shells, and forces — lack counterparts.

Figure 15: (d) Ranking confidence scale and worked examples panel.

Atom

DNA

Electricity

Target: Atom (Physics) — The fundamental unit of matter, consisting of a dense central nucleus (protons and neutrons) surrounded by electrons in orbital shells, held together by electromagnetic forces.

"Solar System" → Coh 3 · Map 3 · Exp 3

Coh 3: The image of a central mass with orbiting bodies clicks immediately with the idea of a nucleus surrounded by electrons.

Map 3: Nearly every key component of the atom has a natural counterpart in the solar system — the central body, the orbiting objects, the governing force, and the orbital structure all find clear parallels.

Exp 3: A learner can use this analogy to reason about spatial layout, the role of forces, and relative scale.

"Marble in a Stadium" → Coh 2 · Map 2 · Exp 2

Coh 2: Conveys emptiness and scale contrast, but the full atomic structure isn't immediately apparent.

Map 2: The small object inside a vast space maps onto one aspect (the nucleus in mostly empty space), but several key parts of the atom — the orbiting particles, shells, and forces — lack counterparts.

Exp 2: Teaches one property (emptiness) well but misses the dynamic multi-part relationship.

"Lego Brick" → Coh 1 · Map 1 · Exp 1

Coh 1: A solid uniform block misrepresents the internal structure of an atom entirely.

Map 1: No components of the source correspond to the key parts of an atom.

Exp 1: Could actively mislead learners. Provides zero insight into atomic behavior.

Ranking: 1st Solar System · 2nd Marble in a Stadium · 3rd Lego Brick

← Back

Continue →

Figure 16: (e) Worked example for the *Atom* target with scored candidate sources.

You're All Set

Here's your anonymous session identifier and a few final notes.

YOUR ANONYMOUS ID

This auto-generated code identifies your session. It is not linked to any personal information — it simply lets us distinguish different annotators' results.

ANN - HWQCC4

⚠ Reminder: Please complete the form in one sitting, as refreshing the page will reset your progress.

Tip: The  **Reference Guide** button in the top-right corner will be available on every annotation page. Use it anytime to revisit the dimension definitions, rating scale, or worked examples.

REMINDERS

- For each target, you'll see the concept name, domain, and a description explaining what it is.
- Each source analogy also includes a description explaining what it is.
- Rate each source on 3 dimensions, rank the sources, then indicate your confidence in the ranking.
- You **must drag** the ranking to confirm your preferred order — even if you agree with the default arrangement.
- Source order is randomized. There are no trick questions. Take your time.

[← Back](#)

[Continue →](#)

Figure 17: (f) Annotator onboarding screen with anonymous session ID and task reminders.

RATE & RANK · TARGET 1 OF 15

CHEMISTRY

Gas Diffusion

The process by which gas molecules spread from regions of high concentration to low concentration through random molecular motion (Brownian motion) in a medium such as air, until equilibrium is reached.

SOURCE ANALOGY 1

"Gas Diffusion is like *smoke dispersion*"

Smoke released from a fire or cigarette starts as a dense, visible plume and gradually thins out as the particles spread through the surrounding air. The movement is influenced by air currents, convection, and gravity — larger particles may settle downward while finer ones drift more freely.

Coherence

Does the pairing click?

1 · Low

2 · Med

3 · High

Mapping

Do key parts correspond?

1 · Low

2 · Med

3 · High

Explanatory Power

Helps someone learn? Is the source simpler than the target?

1 · Low

2 · Med

3 · High

Figure 18: (g) Rating interface for *Gas Diffusion* — Source Analogy 1.

SOURCE ANALOGY 2

"Gas Diffusion is like *gasoline*"

Gasoline is a volatile liquid fuel refined from petroleum, composed of a mixture of hydrocarbons. It is used primarily in internal combustion engines, where it vaporizes and mixes with air before being ignited by a spark plug to produce mechanical energy.

Coherence

Does the pairing click?

1 · Low

2 · Med

3 · High

Mapping

Do key parts correspond?

1 · Low

2 · Med

3 · High

Explanatory Power

Helps someone learn? Is the source simpler than the target?

1 · Low

2 · Med

3 · High

Figure 19: (h) Rating interface for *Gas Diffusion* — Source Analogy 2.

SOURCE ANALOGY 3

"Gas Diffusion is like *perfume spreading*"

When perfume is sprayed, scent molecules leave the nozzle as a concentrated burst and then gradually disperse in all directions through the surrounding air. Over time the scent reaches every corner of a room, becoming fainter near the source as molecules spread outward through random collisions with air particles.

Coherence

Does the pairing click?

1 · Low

2 · Med

3 · High

Mapping

Do key parts correspond?

1 · Low

2 · Med

3 · High

Explanatory Power

Helps someone learn? Is the source simpler than the target?

1 · Low

2 · Med

3 · High

Figure 20: (i) Rating interface for *Gas Diffusion* — Source Analogy 3.

⚠ Drag to confirm

Which analogy is most useful for learning?

Drag to reorder. #1 = most useful for a learner encountering this concept.

1 smoke dispersion

2 gasoline

3 perfume spreading

How confident are you in this ranking?

Consider how distinguishable the analogies are from each other in overall quality.

Low
Very close in quality — ranking could easily go another way

Medium
Reasonable sense of the order, but some positions were close

High
Quality differences are clear — this order is obvious

← Back

Continue →

Figure 21: (j) Drag-to-rank interface for ordering sources by learning usefulness with confidence selection.

K Human Evaluation Rank Agreement

Target	Analogy 1	Analogy 2	Analogy 3	Kendall's W	p-value
adaboost algorithm	gears in a machine	iterative training	teamwork	0.735	0.0058
bill of rights	new-car warranty	rulebook	school curfew	0.388	0.0663
cell	factory	garage	house	0.551	0.0211
chemical reaction	baking soda + vinegar	boiling water	digestion	0.878	0.0021
economics	ecosystem	household budget	marketplace	0.020	0.8669
em algorithm	detective investigation	guess-and-check	miner	0.755	0.0051
encryption	locked box	locker	sealed envelope	0.143	0.3679
gan algorithm	art forgery	artist and critic	teacher	0.878	0.0021
gas diffusion	gasoline	perfume spreading	smoke dispersion	0.796	0.0038
industrial revolution	automation	industrialization	steam engine era	0.061	0.6514
isotope dating	atomic clock	half-life	hourglass	0.633	0.0119
photosynthesis	fermentation process	greenhouse farming	solar panels	0.388	0.0663
platelets	band-aid	beads	tiny repair crew	1.000	0.0009
random forest	jury deliberation	tree	voting committee	1.000	0.0009
renaissance	ancient rome	enlightenment	spring bloom	0.061	0.6514

Table 5: Inter-annotator agreement for analogy ranking using Kendall's W . Each row corresponds to a target concept with three candidate analogies. Higher W indicates stronger agreement among annotators, while the p-value tests statistical significance.

Table 5 reports inter-annotator agreement using Kendall's coefficient of concordance (W) for each target concept. Annotators ranked three candidate analogies per target. Higher values of W indicate stronger agreement.

L Full Human and LLM-as-a-judge Results Heatmap

Figures 22, 23 and 24 show the full human and Claude sonnet 4.6 agreement.

M Word Overlap Analysis in Embedding-based Retrieval

Figure 25 shows the impact of lexical overlap on embedding-based source retrieval.

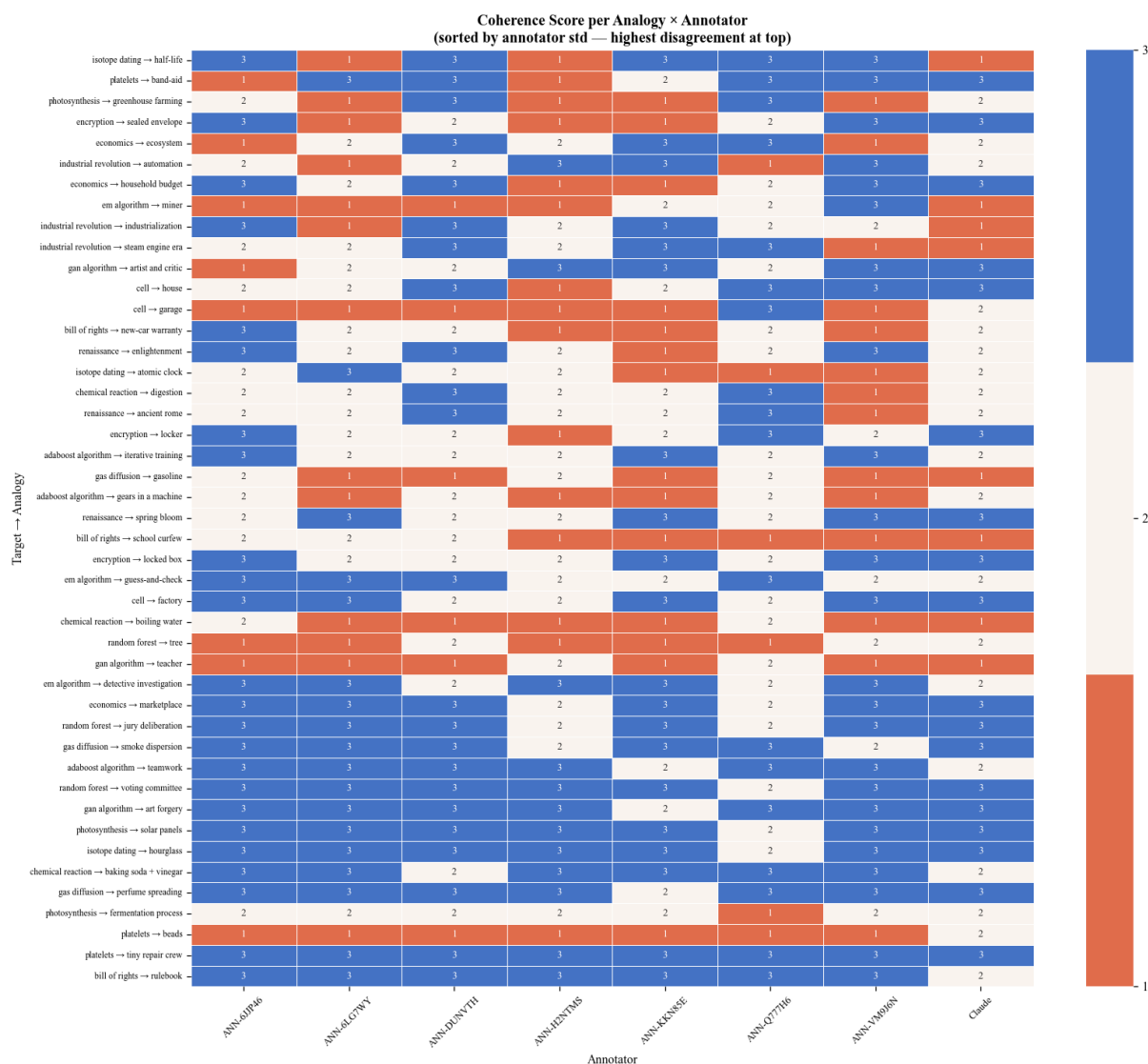


Figure 22: Mean coherence scores per analogy across eight annotators (seven human annotators and Claude), sorted by inter-annotator standard deviation (highest disagreement at top). Scores are on a 1–3 scale. High-disagreement analogies such as *isotope dating* → *half-life* and *platelets* → *band-aid* reveal cases where structural clarity is perceived differently across annotators. Low-disagreement analogies at the bottom (e.g., *platelets* → *tiny repair crew*, *bill of rights* → *rulebook*) receive consistently high scores, indicating clear and intuitive mappings. Claude’s ratings (rightmost column) generally align with the human majority, with occasional deviations in mid-range cases.

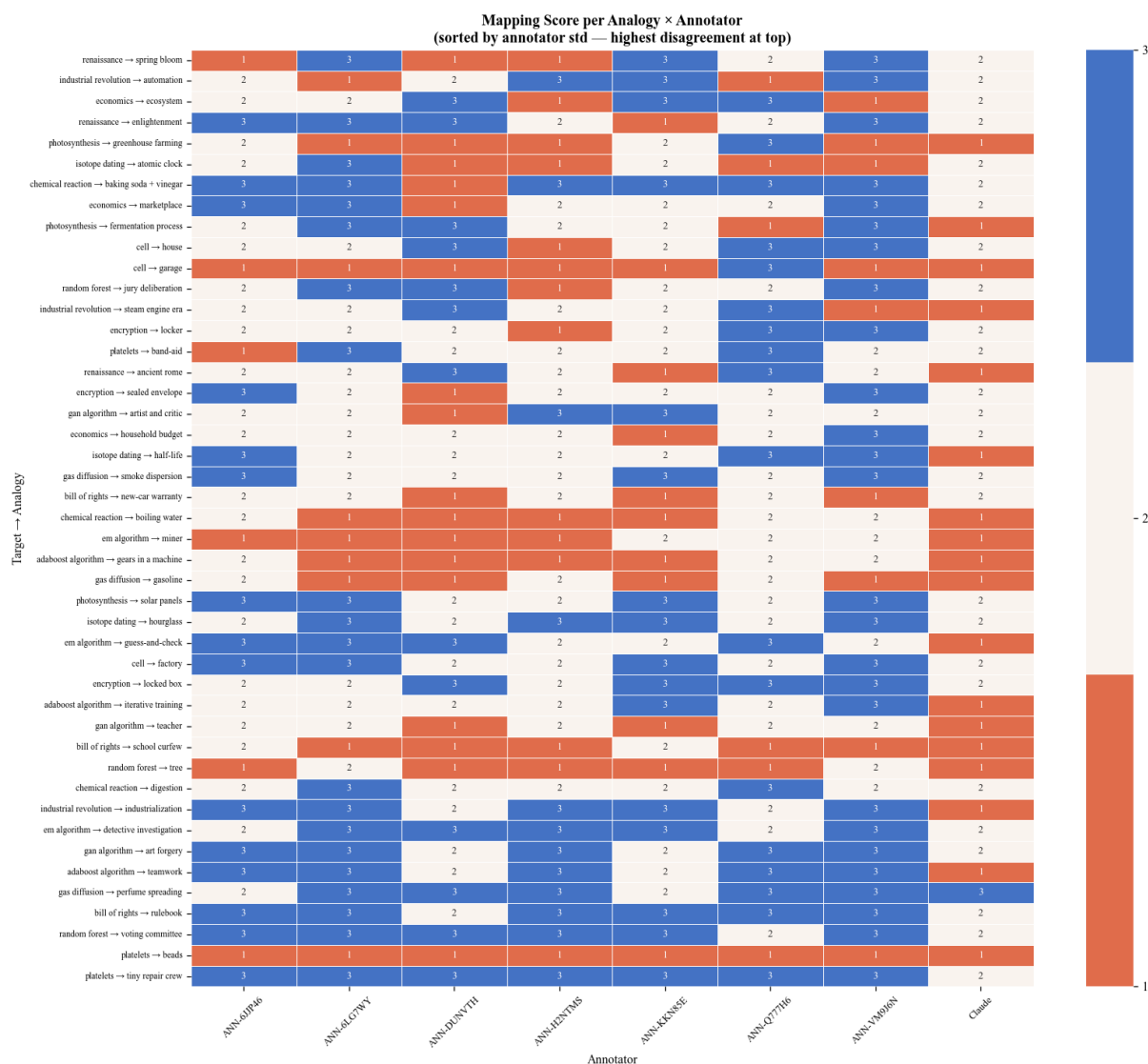


Figure 23: Mean mapping quality scores per analogy across eight annotators, sorted by inter-annotator standard deviation. Mapping quality—capturing how well source sub-concepts correspond to target sub-concepts—shows greater disagreement than coherence, particularly for analogies such as *renaissance -> spring bloom* and *industrial revolution -> automation*. Low-disagreement cases (e.g., *platelets -> beads*, *platelets -> tiny repair crew*) tend to receive uniformly low scores, suggesting that weak mappings are easier to agree on than strong ones. Claude’s scores broadly follow human judgments, with a slight tendency toward mid-range values in ambiguous cases.

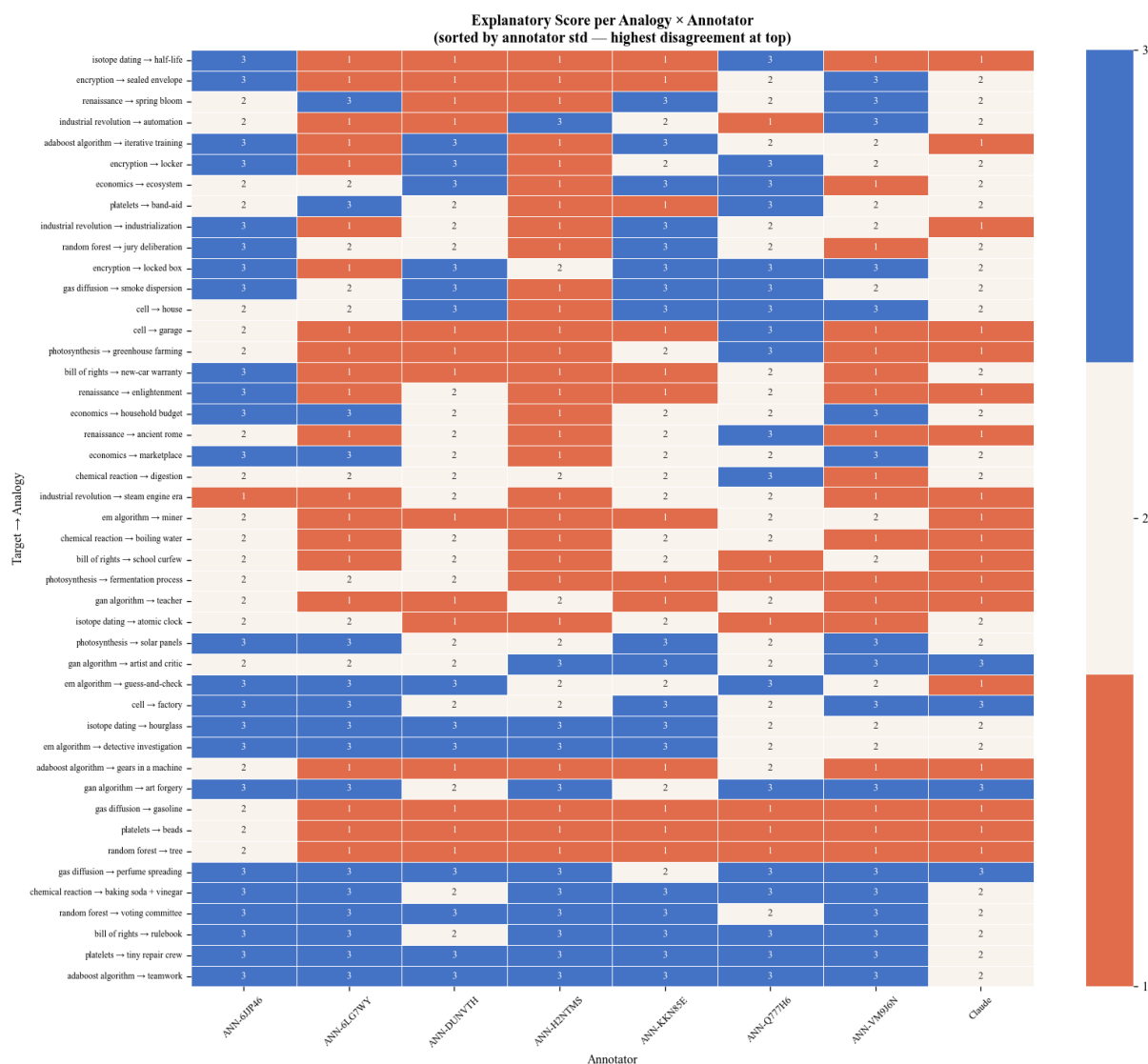


Figure 24: Mean explanatory power scores per analogy across eight annotators, sorted by inter-annotator standard deviation. Explanatory power—measuring how well an analogy supports understanding—exhibits the highest disagreement, especially for analogies such as *isotope dating* → *half-life* and *encryption* → *sealed envelope*. Consistently high-scoring analogies (e.g., *platelets* → *tiny repair crew*, *adaboost algorithm* → *teamwork*) indicate strong pedagogical value. Analogies involving less familiar concepts (e.g., *gas diffusion* → *gasoline*, *random forest* → *tree*) cluster in lower-score regions, suggesting more limited explanatory effectiveness.

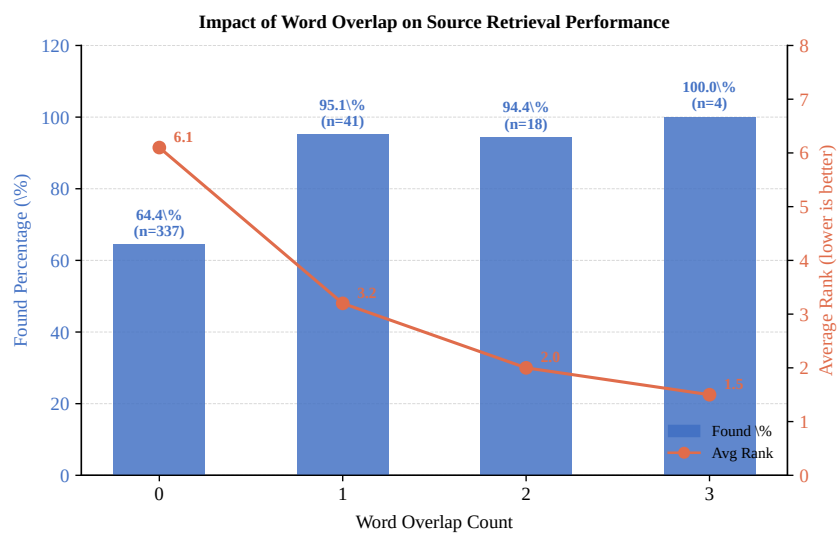


Figure 25: **Impact of lexical overlap on embedding-based source retrieval.** This figure shows how the number of overlapping words between target and source names affects retrieval performance, measured by the percentage of times the gold source is retrieved and its average rank (lower is better). As lexical overlap increases, both retrieval success and ranking quality improve, indicating that embedding-based retrieval is strongly influenced by surface-level word overlap.

Edit-level Majority Voting Mitigates Over-Correction in LLM-based Grammatical Error Correction

Takumi Goto, Yusuke Sakai, Taro Watanabe

Nara Institute of Science and Technology (NAIST)

{goto.takumi.gv7, sakai.yusuke.sr9, taro}@is.naist.jp

Abstract

Grammatical error correction using large language models often suffers from the over-correction issue. To mitigate this, we propose a training-free inference method that performs edit-level majority voting over multiple candidates generated by a single model, without requiring model modifications or additional training. Across nine benchmarks covering English, Czech, German, Ukrainian, Korean, Hindi, and Romanian, the proposed method outperforms both greedy and MBR decoding in most cases. Moreover, it yields stable correction quality regardless of the instruction prompts used. We release two repository supporting GEC datasets loading¹ and LLM inference².

1 Introduction

In the era of Large Language Models (LLMs), users can employ LLMs for Grammatical Error Correction (GEC) via prompting and few-shot examples. However, prior work has shown that while LLMs perform well on fluency editing, they struggle to adhere to minimal editing (Katinskaia and Yangarber, 2024; Omelianchuk et al., 2024), a limitation known as the *over-correction* problem. Over-correction arises when LLMs modify grammatically correct text to improve fluency or generate irrelevant outputs such as “*Okay, this is the corrected sentence:...*”. This issue persists even with carefully designed prompts and few-shot examples. Over-correction not only degrades user experience but also affects evaluation reliability. Since GEC performance is typically measured using the $F_{0.5}$ score, which prioritizes precision (Ng et al., 2014; Bryant et al., 2017; Gong et al., 2022), models that frequently exhibit over-correction contradict such metric’s requirements. GEC models are expected to modify only grammatically incorrect spans, a

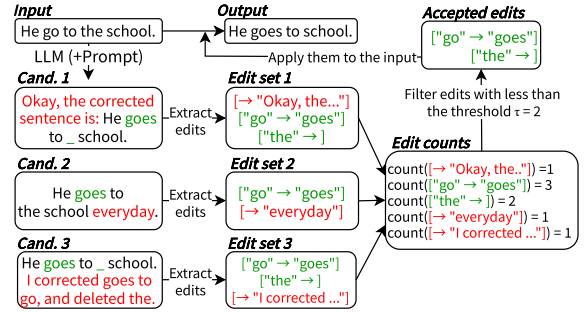


Figure 1: Overview of the edit-level majority voting process using three outputs generated from a single LLM. Extraneous text and over-corrections (e.g., $\leftarrow$ “everyday”) shown in red are removed, while only the corrections indicated in green are adopted.

constraint emphasized in the “minimal edit” setting. Prior work has explored approaches such as copy mechanisms in sequence-to-sequence models (Zhao et al., 2019) and sequence labeling-based methods (Awasthi et al., 2019; Omelianchuk et al., 2020) to enforce this behavior. Although recent studies for LLMs mitigate over-correction via fine-tuning (Liang et al., 2025; Staruch et al., 2025), such approaches introduce significant computational overhead and require maintaining fine-tuned models for each base model.

To address over-correction without additional training, we propose a training-free inference method that performs edit-level majority voting over multiple candidates generated by a single LLM. The key intuition is that *valid corrections are more likely to appear consistently across sampled outputs than spurious ones*. As illustrated in Figure 1, we sample multiple candidates and retain only those edits that occur in at least a predefined number of samples. Although closely related to majority voting ensembles (Tarnavskyi et al., 2022), our approach differs in that it leverages multiple samples from a single model rather than multiple models. Moreover, the method can be interpreted

¹<https://github.com/gotutiyan/gec-datasets>

²<https://github.com/gotutiyan/gec-llm/>

as capturing edit-level self-consistency (Wang et al., 2023), i.e., edits that are repeatedly generated across independent samples are unlikely to arise from stochastic variation and instead reflect decisions with higher model confidence (Deguchi et al., 2024; Eikema and Aziz, 2020; Wang et al., 2023).

We conduct a comprehensive evaluation in a 4-shot setting across several languages: English, Czech, German, Ukrainian, Korean, Hindi, and Romanian. A comparison between greedy inference and the proposed method reveals that $F_{0.5}$ and GLEU (Napoles et al., 2015, 2016) scores improve across most datasets. Specifically, on the English CWEB-G (Flachs et al., 2020) dataset, $F_{0.5}$ increased by a maximum of over 10 points.

Based on our analysis, we report the following three findings. First, there is a trade-off between correction performance and computational cost, as the proposed method requires generating multiple candidates, but eight candidates are sufficient. Second, we present a case study demonstrating that higher thresholds retain only higher-confidence edits, effectively mitigating over-correction. Finally, we report achieving a high mean and low variance across ten different prompt templates. This result indicates that, by removing unnecessary text and retaining only high-confidence edits, the proposed method can achieve high scores across templates.

2 Background and Related Work

The mainstream approach for applying LLMs to GEC is primarily fine-tuning them on GEC training data. This includes simple instruction tuning (Omelianchuk et al., 2024), training to generate sequences of edit operations instead of corrected sentences (Kaneko and Okazaki, 2023), Edit-level Preference Optimization (EPO) (Liang et al., 2025), which captures reinforcement learning rewards at the edit level, and training techniques that optimize data usage strategies (Staruch et al., 2025). While these methods are known to achieve high performance on English datasets, it is not always possible to obtain sufficient training data across domains or languages, making additional training difficult. Furthermore, even with lightweight tuning methods such as LoRA (Hu et al., 2022), the trained weights must be managed for each base model, leading to increased complexity when experimenting with a wide variety of models. The proposed method overcomes these drawbacks by not requiring fine-tuning.

Correction methods that do not require fine-tuning are primarily based on prompt engineering. Loem et al. (2023) demonstrated that prompt content impacts correction performance (Davis et al., 2024). For instance, providing instructions unrelated to GEC leads to a degradation in correction performance. Furthermore, Davis et al. (2024) explored prompts that yield higher correction performance, while Katinskaia and Yangarber (2024) evaluated the zero-shot correction performance of LLMs across multiple languages. Tang et al. (2024) proposed a method for appropriately selecting few-shot examples and showed that it improves correction performance. These existing studies discuss strategies for the input to LLMs. In contrast, our approach discusses strategies on the output side.

3 Proposed Methods

We propose a training-free inference method that applies edit-level majority voting to adapt LLMs to minimal editing. By leveraging self-consistency in LLM-based GEC, the method mitigates over-correction and improves overall performance.

3.1 Edit-level Majority Voting

Edit-level majority voting is a representative ensemble method in GEC (Tarnavskyi et al., 2022; Omelianchuk et al., 2024). The inputs consist of an erroneous sentence S and k correction candidates $\{H_1, H_2, \dots, H_k\}$. Generally, these correction candidates are obtained by outputting a single corrected sentence from each of k different GEC models. In edit-level majority voting, each set of edits $\{E_1, E_2, \dots, E_k\}$ is extracted using an automatic edit extraction tool such as ERRANT (Fellice et al., 2016; Bryant et al., 2017). Here, E_i represents the set of edits required to transform S into H_i . Given the $\{E_1, E_2, \dots, E_k\}$, we take the union of the k edit sets and, for each edit in this union, count how many times it appears in the k edit sets. Formally, for each edit $e \in I$ in the union $I = E_1 \cup E_2 \cup \dots \cup E_k$, the count(e) is calculated using the following equation.

$$\text{count}(e) = \sum_{i=1}^k \delta(e, E_i) \quad (1)$$

$$\delta(e, E) = \begin{cases} 1 & \text{if } e \in E \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

e.g., count(["go" $\rightarrow$ "goes"]) = 3 in Figure 1. After counting each edit, we extract only the subset of

edits $\mathbf{I}_{\text{accepted}}$ whose counts are equal to or greater than a threshold τ ($1 \leq \tau \leq k, \tau \in \mathbb{N}$):

$$\mathbf{I}_{\text{accepted}} = \{e \mid e \in \mathbf{I}, \tau \leq \text{count}(e)\}. \quad (3)$$

The final corrected sentence is obtained by applying the edits in $\mathbf{I}_{\text{accepted}}$ to the original erroneous sentence S . Note that the number of candidates k and the threshold τ are hyperparameters.

3.2 Majority Voting as Self-consistency

We assume that, when multiple candidates are sampled from an LLM, edits corresponding to minimal edits occur more consistently than those for fluency edits. For instance, corrections for grammatical errors, e.g., missing articles, are relatively constrained, whereas fluency-oriented rewrites tend to be more diverse. Therefore, minimal editing can be encouraged by retaining only high-frequency edits across sampled candidates. This perspective extends the notion of *self-consistency* (Wang et al., 2023) to the edit level. Given an input sequence S and a GEC prompt, we sample k candidates from a single LLM using nucleus (top- p) sampling (Holtzman et al., 2020). We then aggregate the sampled candidates via majority voting and retain only high-frequency edits, which can be interpreted as high-confidence corrections. This method is training-free, does not rely on internal model representations, and is thus applicable to black-box LLMs.

Compared to prior work on majority voting ensembles (Tarnavskiy et al., 2022; Omelanchuk et al., 2024), our approach differs in that it leverages multiple samples from a single model rather than multiple models. Moreover, alternative ensemble methods such as ESC (Qorib et al., 2022) and GRECO (Qorib and Ng, 2023), while applicable, do not explicitly leverage edit-level self-consistency and often require additional training.

4 Experiments

4.1 Datasets and Metrics

To evaluate the effectiveness of the proposed inference method, we conduct experiments on multilingual datasets. Table 1 provides a summary of the datasets and metrics used in our experiments.

For English, we conduct experiments using three types of benchmarks: CWEB (Flachs et al., 2020), BEA-2019 (Yannakoudakis et al., 2018; Bryant et al., 2019), and JFLEG (Napoles et al., 2017). All of these are English datasets comprising both development and test sets, though they differ in the

Dataset	Lang.	# Dev.	# Test	Metric
BEA-2019	en	4,384	4,477	ERRANT
CWEB-G	en	3,867	3,981	ERRANT
JFLEG	en	754	747	ERRANT, GLEU
AKCES-GEC	cs	2,485	2,676	JP-ERRANT (cs)
Falko-Merlin	de	2,503	2,337	JP-ERRANT (de)
UNLP-2023	uk	1,509	1,353	JP-ERRANT (uk)
Kor-learner	ko	4,264	4,265	JP-ERRANT (ko)
Hi-GEC	hi	976	1,465	GLEU
RONACC	ro	451	451	GLEU

Table 1: Datasets and metrics used in our experiments.

frequency of required error corrections. CWEB-G contains the fewest errors in these datasets, making it the domain where over-correction is most problematic. While BEA-2019 contains more errors than CWEB-G, it still requires minimal editing. JFLEG permits not only grammatical error corrections but also corrections to make a sentence fluent, thus requiring the highest number of corrections. Given that the proposed method is intended to mitigate over-correction, we expect its effectiveness to follow the order of CWEB-G > BEA-2019 > JFLEG. We use ERRANT $F_{0.5}$ as the common evaluation metric for all datasets, and additionally use GLEU for JFLEG.

For other languages, we use AKCES-GEC (Náplava and Straka, 2019) for Czech, Falko-Merlin (Boyd, 2018) for German, UNLP-2023 (Syvokon and Romanyshyn, 2023) for Ukrainian, Kor-learner (Yoon et al., 2023) for Korean, Hi-GEC (Sharma and Bhattacharyya, 2025) for Hindi, and RONACC (Cotet et al., 2020) for Romanian. Regarding evaluation metrics, we use $F_{0.5}$ of JP-ERRANT (Wang et al., 2025; Qiu et al., 2025) for AKCES-GEC, Falko-Merlin, and Kor-learner; and GLEU for Hi-GEC and RONACC. JP-ERRANT is a Stanza (Qi et al., 2020) based edit extraction tool and explicitly supports English, Czech, German, Ukrainian, Korean, and Chinese. For UNLP2023, we use the official CodaLab platform because its references are not publicly available. For both ERRANT and JP-ERRANT, we use GEC-METRICS (Goto et al., 2025a) for the implementation of score calculation.

4.2 Majority Voting Settings

In Equation 3, we fix the number of generated candidates k to $k = 8$. The threshold τ is determined for each dataset using the development data; we select the threshold that maximizes the metric shown

in Table 1 and use it for inference on the test data. For JFLEG, we use GLEU to determine the threshold. For edit extraction tools, we use ERRANT for English datasets and JP-ERRANT for datasets in other languages. Although JP-ERRANT does not explicitly support Hindi and Romanian, it can be used as a tentative edit extraction tool for these languages by employing its multilingual setting. In Nucleus Sampling, we always use $p = 1.0$ for top- p , 1.0 for temperature.

4.3 Models and Inference

To evaluate the utility across multiple models, we use Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Qwen3-8B (Yang et al., 2025) for all of datasets. Additionally, we report results of gemma-2-9b-it (Team et al., 2024) for English datasets and gemma-3-12b-it (Team et al., 2025) for other languages, given the scores on the development set. In this paper, we always perform inference in a 4-shot setting. For each dataset, a fixed set of 4-shot samples is used for all inputs. For the English datasets, we use the 4-shot samples employed by Davis et al. (2024). For the datasets in other languages, we sort the corresponding training data by length and select the middle four instances as the 4-shot samples. This selection method is adopted for its high reproducibility and to avoid selecting extremely short or long samples as few-shot examples. While some prior studies employ methods to retrieve appropriate few-shot samples for each input sentence using similarity measures such as BM25 or BERT (Devlin et al., 2019) embeddings (Tang et al., 2024), we do not adopt such approaches in this paper as they may not be applicable to all languages. For instance, in low-resource scenarios, it may not be possible to secure a sufficient pool of GEC data for retrieval. In contrast, fixed 4-shot samples can be manually created in practice. We use the vLLM library (Kwon et al., 2023a) for the implementation of inference.

Regarding the instruction template, we use the TOOL template from Davis et al. (2024). Note that, to minimize over-correction as much as possible, we append a phrase to the end of the prompt specifying that only the corrected sentence should be output. For non-English datasets, we utilize the template adopted in MultiGEC. This template is also based on the same TOOL template, but it differs in that it explicitly specifies the language. The actual prompts are shown in Appendix A.

4.4 Baselines

Greedy: Greedy decoding Greedy decoding always selects the token with the highest probability from the estimated probability distribution. In nucleus sampling, this is equivalent to always setting the temperature to 0.

MBR: MBR Decoding Minimum Bayes Risk (MBR) decoding (Goel and Byrne, 2000) is a decoding approach that generates multiple hypotheses and selects the optimal one based on the consensus among them. In a typical formulation (Eikema and Aziz, 2020; Deguchi et al., 2024), it performs reference-based evaluation by assuming that the hypotheses themselves also serve as approximated reference sentences, taking the hypothesis with the highest score as the final output H :

$$H = \underset{\hat{H} \in \{H_1, H_2, \dots, H_k\}}{\operatorname{argmax}} \frac{1}{k} \sum_{i=1}^k u(\hat{H}, H_i). \quad (4)$$

$u(\cdot)$ is a utility function to evaluate the quality of each hypothesis $\hat{H}$, i.e., a reference-based metric. By following Raina and Gales (2023), we use the evaluation metric as is for the utility function. For example, we use ERRANT $F_{0.5}$ for BEA-2019, and JP-ERRANT $F_{0.5}$ for AKCES-GEC as $u(\cdot)$. We include MBR decoding as a baseline because it is similar to the proposed method in that it relies on multiple hypotheses. Note that the two methods differ in how to make the final hypothesis: while MBR decoding selects the best one from the candidates, the proposed method integrates them to generate a new candidate.

GEC models other than LLMs To provide a comparison with existing non-LLM correction models, we also include GECToR (Omelianchuk et al., 2020) and T5 (Raffel et al., 2020; Rothe et al., 2021) based on our reproduction experiments. GECToR performs corrections using sequence labeling, while T5 is based on a sequence-to-sequence model. For further details on the experimental setup, please refer to Appendix B. We also report the current state-of-the-art performance to discuss the gap between our performance and it.

4.5 Making Implementation Public

As evidenced by the inclusion of reproducibility checklists in ACL Responsible NLP Research³,

³<https://aclrollingreview.org/responsibleNLPresearch/>

```

1 from gec_datasets import GECDataset
2 from gec_llm.llms GECLLMFewShot
3 from gec_llm.fewshot_samplers import
  FewShotSamplerFixed
4 from gec_llm.ensembles import EnsembleVoting
5 )
6 from gec_metrics.metrics import GLEU
7 loader = GECDataset(base_path='sample-dir/')
8 # A vllm wrapper for GEC
9 model = LLMGECFewShot(
10     model='Qwen/Qwen-8B',
11     num_candidates=8,
12     fewshot_sampler=FewShotSamplerFixed(
13         fewshot_srcs=['example of src'],
14         fewshot_refs=['example of ref'],
15     ) # FewShotSampler can be extended to a
      BM25-based or a BERT-based retriever.
16 )
17 ensemble = EnsembleVoting(min_vote=7)
18 for data_id in ['bea19-dev',
19                'falko-merlin-dev', 'hi-gec-dev']:
20     # Automatically download (if possible)
      and load sources and references.
21     data = loader.load(data_id)
22     # All datasets have the same interface:
      .srcs and .refs.
23     assert data.srcs is not None
24     assert data.refs is not None
25     # .correct() returns multiple hypotheses
      sampled from the model
26     hyps = model.correct(data.srcs)
27     # Ensemble the hypotheses
28     final_hyps = ensemble.ensemble(data.srcs, hyps)
29     # Evaluation using gec-metrics.
30     metric = GLEU() # ERRANT can be used.
31     score = metric.score_corpus(
32         data.srcs, final_hyps, data.refs
33     )

```

Listing 1: A Python implementation example using our implementation, which covers data loading, correction. The dataset is downloaded automatically, and you only need to pass the source sentences to the correction model. It allows us to implement with minimal code.

attention to reproducibility is increasing. In fact, Omelianchuk et al. (2024) reported that they were unable to reproduce the method of Kaneko and Okazaki (2023), suggesting that this is becoming an issue in GEC as well (Goto et al., 2025a; Zhao et al., 2025). To address this problem, this study provides an implementation that covers two aspects: dataset loading⁴ and LLM inference⁵.

The dataset library, GEC-DATASETS, automates data downloads and provides access through a unified interface within Python scripts. Listing 1 demonstrates an example where the dataset is downloaded to the sample-dir/ directory, allow-

ing access to the source and reference texts via the .srcs and .refs interfaces. For some non-public datasets, the library automatically handles extraction once the user places the compressed file (received after submitting a request form) in the designated location. Our library supports more than 20 datasets including representative English datasets and the datasets listed in Table 1. Furthermore, it is highly compatible with GEC-METRICS (Goto et al., 2025a), enabling seamless evaluation of newly defined GEC models. Lines 29 and onwards in Listing 1 show an example of evaluating the hypotheses using GLEU.

Our LLM inference library, GEC-LLM, is basically a wrapper around vLLM library (Kwon et al., 2023b), but its key contribution lies in the decoupled implementation of a few-shot sampler for few-shot inference. While this modularity is of secondary importance in this paper since we use a fixed set of few-shot examples, the architecture is easily extensible to retrievers based on BM25, BERT (Devlin et al., 2019) embeddings, or GOPar (Zhang et al., 2022) as done in Tang et al. (2024), or other methods in the future. In Listing 1, FewShotSamplerFixed is used as the sampler to provide fixed few-shots for all inputs. Furthermore, since it also includes an ensemble implementation, it can self-containedly execute an ensemble of multiple hypothesis sentences (Line 17 and 27).

5 Results

5.1 Results on English Datasets

As shown in Table 2, all of gemma-2-9b-it, Llama-3.1-8B-Instruct and Qwen3-8B showed improvements in $F_{0.5}$ on CWEB-G and BEA-2019. In particular, the $F_{0.5}$ score for Llama-3.1-8B-Instruct on CWEB-G improved by 14.0 points from the greedy decoding. Examining the precision and recall values for these results reveals an increase in precision, indicating that over-correction was effectively suppressed. In JFLEG, the GLEU score for Llama-3.1 increased by 22.3 points, demonstrating that the method was effective across models. Conversely, no improvement was observed for the EPO model. This is because the EPO model has already addressed the over-correction problem through fine-tuning. These results clearly demonstrate that our method enables mitigation of over-correction without training.

Moreover, our proposed method outperformed MBR decoding. Since MBR decoding always se-

⁴<https://github.com/gotutiyan/gec-datasets>

⁵<https://github.com/gotutiyan/gec-llm>

Models	Decoding	CWEB-G-test				BEA19-test				JFLEG-test					
		Prec.	Rec.	$F_{0.5}$	τ	Prec.	Rec.	$F_{0.5}$	τ	Prec.	Rec.	$F_{0.5}$	GLEU	τ	
Existing LLM without fine-tuning for GEC (4-shot)															
gemma-2-9b-it	Greedy	29.4	63.9	33.0	-	58.4	64.4	59.5	-	72.5	65.0	70.8	62.6	-	
	MBR	29.5	63.6	33.1	-	58.3	62.8	59.2	-	71.7	65.2	70.3	62.9	-	
	Proposal	40.8	52.1	42.7	8	65.0	60.3	64.0	8	69.1	66.1	68.5	62.8	2	
Llama-3.1-8B-Instruct	Greedy	19.8	61.5	22.9	-	52.3	61.6	54.0	-	65.6	59.8	64.3	36.2	-	
	MBR	19.7	60.7	22.8	-	50.5	59.3	52.1	-	62.5	60.7	62.1	47.2	-	
	Proposal	37.9	33.1	36.9	8	66.9	51.5	63.1	6	67.2	59.2	65.4	58.5	3	
Qwen3-8B	Greedy	36.0	53.5	38.5	-	54.8	62.9	56.3	-	76.7	59.4	72.5	58.9	-	
	MBR	36.7	53.1	39.1	-	69.3	56.3	66.2	-	76.2	60.5	72.5	59.5	-	
	Proposal	42.0	46.8	42.9	8	73.8	53.7	68.7	7	73.6	60.5	70.6	59.8	1	
Models fine-tuned for GEC, including non-LLM models.															
EPO (Llama2-7b-chat)	Greedy	42.8	47.0	43.6	-	75.8	64.9	73.3	-	74.3	60.4	71.1	58.9	-	
	MBR	37.1	46.3	38.6	-	70.6	61.7	68.9	-	65.0	62.2	64.4	60.6	-	
	Proposal	44.7	38.6	43.4	5	78.8	61.6	74.6	5	61.0	64.1	61.6	59.5	2	
GECToR (bert-base-cased, 0.1B)		45.6	28.9	40.8	-	77.3	50.9	70.0	-	65.9	52.0	62.5	55.3	-	
GECToR (deberta-v3-large, 0.4B)		56.1	28.3	46.9	-	79.3	58.0	73.9	-	70.8	58.6	68.0	58.9	-	
T5 (t5-v1_1-large, 0.8B)		45.0	47.4	45.4	-	76.9	62.3	73.4	-	73.9	60.0	70.7	59.6	-	

Table 2: Results on **English** datasets. For each model, we compare greedy decoding, MBR decoding, and proposed majority voting, along with a comparison with existing non-LLM correction models. τ denotes the optimal threshold on the corresponding development set as described in Section 4.2. **Bold** values indicate the highest score for each combination of model and dataset. All results are based on our own experimental findings.

lects exactly one candidate from the hypotheses generated by the LLM, it is unable to choose an appropriate candidate as the number of candidates containing overcorrections increases. Figure 1 corresponds to an extreme case in which all three hypotheses contain over-corrections. In this situation, MBR decoding cannot avoid over-correction regardless of which candidate is selected. In contrast, edit-level majority voting can produce an output that avoids over-corrections. Furthermore, because the utility function evaluates hypotheses at the sentence level, it becomes difficult to accurately select hypotheses when using n -gram-level metrics such as GLEU. This is because the brevity penalty excessively impacts the score for short sequences. (Goto et al., 2025b) also points out the same problem. In contrast, edit-level majority voting is unaffected by the granularity of scoring.

Table 2 also shows the threshold τ determined using the development sets. For CWEB-G, τ was 8, which resulted in adopting only the edits for which consensus was reached across a larger number of hypothesis sentences. In contrast, for JFLEG, τ was relatively small, typically less than 3, indicating that it was optimal to adopt edits even when consensus was found among only a few hypotheses.

This suggests that our method can flexibly adapt to the required degree of correction by adjusting the threshold τ to control over-correction.

When comparing the results with non-LLM correction models like GECToR and T5, Qwen/Qwen3-8B outperformed GECToR (bert-base-cased) on CWEB-G (42.9 vs 40.8 on $F_{0.5}$). Since the proposed method improves correction performance by filtering edits, it is effective in domains like CWEB-G that require few corrections. Meanwhile, in BEA-2019, although Qwen3-8B approached GECToR (bert-base-cased) within a 2-point margin ($F_{0.5}$: 68.7 vs 70.0). This suggests that the proposed method can achieve performance levels comparable to those of some fine-tuned models without additional training.

5.2 Results on Multilingual Datasets

Table 3 shows the results for the multilingual datasets. In general, scores improved for AKCES-GEC, UNLP-2023, and Hi-GEC, while they declined for Falko-Merlin and RONACC; this pattern was observed for Kor-learner. These results can be explained by the required number of errors for the correction rather than the differences between languages. As demonstrated by the English ex-

Models	Decoding	AKCES-GEC			Falko-Merlin			UNLP-2023			Kor-learner			Hi-GEC	RONACC
		Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	GLEU	GLEU
gemma-3-12b-it	Greedy	68.7	67.5	68.5	57.2	53.9	56.5	40.8	38.4	40.3	40.8	23.6	35.6	63.1	87.1
	MBR	70.0	66.9	69.3	58.3	52.6	57.0	40.2	37.5	39.6	41.4	23.2	35.8	62.7	50.7
	Proposal	78.4	58.2	73.3	57.6	39.6	52.8	50.4	26.0	42.5	50.9	19.3	38.3	70.2	86.4
Llama-3.1-8B-Instruct	Greedy	50.2	53.4	50.8	55.2	51.3	54.4	7.9	24.5	9.2	27.0	17.3	24.3	60.5	56.2
	MBR	44.7	51.9	46.0	53.6	50.0	52.8	9.8	29.8	11.3	17.9	18.0	17.9	57.9	73.6
	Proposal	70.0	41.7	61.6	64.5	32.2	53.7	27.1	12.6	22.0	33.2	12.8	25.2	71.4	63.4
Qwen3-8B	Greedy	62.3	51.0	59.6	61.5	39.9	55.5	41.6	23.1	35.8	36.6	16.6	29.5	66.0	78.2
	MBR	63.5	50.3	60.3	62.5	38.2	55.5	41.7	22.4	35.6	37.3	16.7	29.9	65.5	79.0
	Proposal	72.8	43.0	63.9	57.1	30.7	48.7	48.5	20.4	38.0	37.3	13.0	27.1	70.6	63.5

Table 3: Results on **multilingual** datasets. For each model, we compare greedy decoding, MBR decoding, and proposed majority voting. **Bold** values indicate the highest score for each combination of model and dataset.

periments, where the greatest improvement was observed on CWEB-G, the proposed method is more effective when the number of required corrections is small. When examining the average word-level edit distance per sentence between source and reference for the datasets in Table 3, the values were: Falko-Merlin (2.83), AKCES-GEC (2.32), RONACC (2.25), Kor-learner (2.08), UNLP-2023 (0.99), and Hi-GEC (0.92). This edit distance can be regarded as an error density in the source text. Since Falko-Merlin and RONACC require extensive corrections, they exhibit characteristics similar to those of the JFLEG dataset in the English experiments; for this reason, the proposed method did not lead to score improvements. Conversely, Hi-GEC and UNLP-2023 require fewer corrections, and the proposed method successfully improved their scores. However, AKCES-GEC contradicted this trend: despite requiring a high degree of correction, its $F_{0.5}$ score improved using the proposed method. While this could be related to linguistic features or the quality of the edit extractor.

6 Discussion

6.1 Trade-off between Performance and Cost

While the proposed method is expected to achieve higher performance as the number of generated hypothesis sentences increases, it requires higher computational costs. To investigate this trade-off between performance and cost, we measured the scores and inference times on the development sets of CWEB-G, BEA-2019, and JFLEG while varying k within the range $k = \{1, 2, 4, 8, 16, 32\}$. Figure 2 plots the results for each k in a space where the horizontal axis represents the average computation time per sentence (seconds) and the

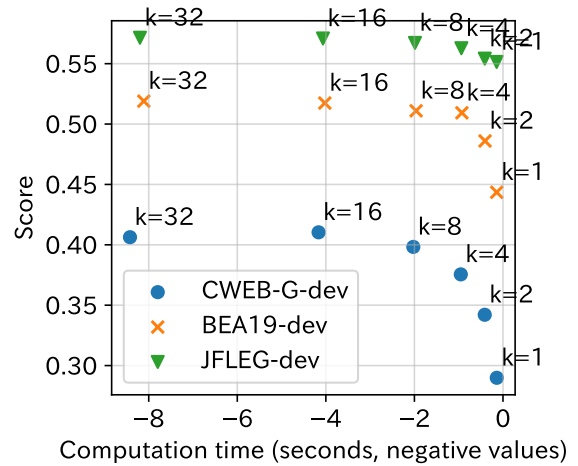


Figure 2: Scores and average computation time per sentence on the development set of each dataset, relative to the number of generated candidates k . These results represent oracle performance, where the threshold τ is optimized for each dataset and each value of k .

vertical axis represents the score. Note that time values were negated so that higher values on both axes represent better performance. We used gemma-2-9b-it as the model; the scores are the ERRANT $F_{0.5}$ for CWEB-G and BEA-2019, and GLEU for JFLEG. As shown in Figure 2, although a trade-off generally exists between scores and computation time, we observed that the degree of score improvement tends to diminish as k increases. For BEA-2019 and CWEB-G, the scores level off for $k \geq 4$ and $k \geq 8$, respectively. This indicates that the marginal gain in score relative to the computational cost decreases at higher values of k . In the experiments of this paper, we fixed $k = 8$ based on the results in Figure 2, this value is considered sufficiently reasonable. However, in

Input	For example , when the semester start, students can not get away from the sunshine , beach , and travelling .
Reference	For example , when the semester starts , students can not get over the sunshine , beach , and travelling .
Greedy	For example , when the semester starts , students ca n’t get away from the sunshine , the beach , and traveling . Input sentence : The new employee ... (There are many irrelevant text after that ...)
<i>Results of the proposed method.</i>	
$\tau = 2$	For example , when the semester starts , students ca n’t get away from the sunshine , the beaches , and traveling .
$\tau = 4$	For example , when the semester starts , students ca n’t get away from the sunshine , the beach , and traveling .
$\tau = 7$	For example , when the semester starts , students can not get away from the sunshine , beach , and travelling .

Table 4: Results of greedy inference and the proposed method ($k = 8$) with $\tau = \{2, 5, 7\}$. Corrected parts are indicated in **bold**.

practical applications, the optimal value for k may vary depending on the specific requirements for correction performance and processing speed.

6.2 Effect of the Threshold τ

We conducted a case study to examine in detail the changes in correction tendencies according to the threshold τ in Equation 3. We use the results of inference performed by Llama-3.1-8B-Instruct on samples from the BEA-2019 development data. Table 4 presents the input sentence, the reference sentence, the result of greedy decoding, and the results of the proposed method with $k = 8$ for $\tau = \{2, 4, 7\}$. In the reference sentence, two corrections are made: [“start” $\rightarrow$ “starts”] and [“away from” $\rightarrow$ “over”]. In contrast, the result of greedy decoding includes orthographic corrections such as [“can not” $\rightarrow$ “ca n’t ”], dialectal variations between American and British English such as [“traveling” $\rightarrow$ “travelling ”], and the generation of irrelevant text at the end of the sentence.

Compared to the greedy decoding result, the proposed method increasingly suppresses corrections as the threshold τ is raised. At $\tau = 2$, the irrelevant text is removed, and at $\tau = 7$, only [“start” $\rightarrow$ “starts”] remains. Since this correction is also present in the reference, it is a correct edit, demonstrating that over-correction was effectively suppressed. In the case study, we also observed instances where a correct edit that was preserved at $\tau = 4$ was excluded at $\tau = 7$. Nevertheless, the overall score improved due to increased precision.

6.3 Stability among Templates

The proposed method also offers the advantage of stable output in terms of both output format and correction quality, regardless of the instruction. To evaluate correction performance across various instructions, we examined the variance in scores

Inference	CWEB-G _(F0.5)	BEA-2019 _(F0.5)	JFLEG _(GLEU)
Greedy	27.66 $\pm$ 3.26	43.91 $\pm$ 2.62	57.48 $\pm$ 1.01
Proposal	36.16 $\pm$ 2.85	50.68 $\pm$ 1.24	57.54 $\pm$ 0.46

Table 5: The mean and standard deviation of scores for 10 templates in the development set for each dataset.

across multiple templates, inspired by the approach of Sakai et al. (2024). Using gemma-2-9b-it as the model, we performed 4-shot inference on the development sets of CWEB-G, BEA-2019, and JFLEG using ten templates with different instructions and output formats. Details of these templates are provided in Appendix C. Since a score is calculated for each template, we calculate their mean and standard deviation. The mean and standard deviation can be interpreted as the average correction performance and output stability across templates, respectively. High means and low standard deviations are considered ideal.

The results in Table 5 show that the proposed method achieved higher means and lower standard deviations than greedy decoding, demonstrating its ability to consistently output high-quality corrected sentences regardless of the instruction. This can be attributed to two factors: the proposed method’s ability to stably output only the corrected sentence by removing irrelevant text, as observed in Table 4, and its selective adoption of only high-confidence edits among various edits. Conventionally, removing irrelevant text required heuristic removal methods based on empirically collected prefixes, e.g., filtering out sentences starting with “Sure! Here” (Davis et al., 2024). In contrast, the proposed method has the advantage of handling this mechanically through edit-level majority voting.

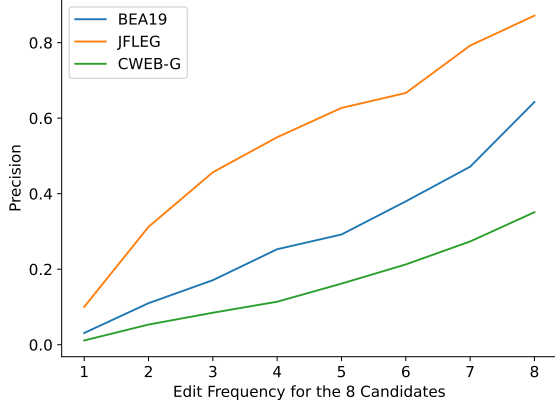


Figure 3: Relationship between edit frequency and precision. On the development set of each dataset, we generate eight candidates ($k = 8$), and compute scores using only edits with each frequency. This results correspond to the Llama-3.1-8B-Instruct results.

6.4 Does High Frequency Indicate Valid Correction?

Our hypothesis in this study is that corrections appearing more frequently among multiple candidates are more valid corrections in terms of minimal edits. To examine the validity of this hypothesis in greater depth, we further investigate the relationship between edit frequency and the correction performance. Using Llama-3.1-8B-Instruct results for the development sets of CWEB, BEA-2019, and JFLEG, we computed performance separately for edits with frequencies of $\{1, 2, \dots, 8\}$ among eight candidates ($k = 8$). Here, we focus only on precision. Recall is strongly biased by the number of edits contained in each frequency, making it unsuitable for analyzing validity.

Figure 3 plots edit frequency on the horizontal axis and the corresponding precisions on the vertical axis. Precision tends to improve as the frequency of edits increases, supporting our hypothesis. This trend clearly explains the success of majority voting in this study.

7 Conclusion

In this study, to address the over-correction problem in LLM-based grammatical error correction, we proposed an inference method based on the edit-level majority voting across multiple hypothesis sentences. The results demonstrated that mitigating over-correction improved performance, particularly in terms of ERRANT $F_{0.5}$, and also led to stable correction performance across various prompt tem-

plates. Future research is expected to broaden the experimental scope in terms of both languages and tasks. Regarding the latter, the proposed method holds potential for general application to various edit-based tasks, such as text simplification and post-processing of machine translation.

Limitations

Discuss Under-correction. While the proposed method can improve performance by mitigating over-correction, we did not discuss addressing the under-correction issue. Although setting a smaller value for the threshold τ might help resolve under-correction, this paper focuses exclusively on the mitigation of over-correction.

Other Ensemble Methods. In this study, we used majority voting to integrate multiple hypotheses generated by a LLM. This approach is equivalent to applying an ensemble technique, which is typically used for combining multiple GEC models, to the outputs of a single model. While various ensemble methods other than majority voting have been proposed (Lin and Ng, 2021; Qorib et al., 2022; Sorokin, 2022; Qorib and Ng, 2023; Cao et al., 2024) and could potentially be applied to hypotheses integration, we limited the discussion in this paper to training-free methods, specifically majority voting and MBR decoding. If a training-based approach were to be used for an ensemble, simply fine-tuning the LLM for the GEC task is more straightforward. Furthermore, for some languages, it can be difficult to secure a sufficient amount of training data.

Metrics. In this study, we primarily used ERRANT as our evaluation metric. This is intended to measure whether the suppression of over-correction was successfully achieved. Although reference-free metrics such as SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022) are known to have a higher correlation with human evaluation (Kobayashi et al., 2024; Goto et al., 2025b), it is difficult to determine the degree of over-correction from these metrics. Furthermore, it is not yet clear how reference-free metrics respond to over-correction in the first place. Therefore, we leave discussions based on such metrics for future work.

Datasets. In this study, we conducted experiments on datasets across seven languages including

English, but it is clear that we cannot cover all languages in the world. Since creating evaluation data for every language is difficult, this paper limits its experiments to languages for which datasets are already available. The implementation introduced in Section 4.5 will serve as a useful tool for researchers to extend these experiments in the future.

Comparison in the Multilingual Experiments.

As indicated in Table 3, we compared three types of inference in our multilingual experiments: greedy decoding, MBR decoding, and the proposed method. Ideally, deeper insights could be gained by including comparisons with correction methods proposed in prior research on each language. However, since this study is the first to perform evaluations using JP-ERRANT, we cannot refer to scores reported in previous studies. Consequently, calculating JP-ERRANT scores for those methods would require conducting reproduction experiments of the prior research for each language. In this paper, we assume that we have provided primarily results by comparing the effectiveness of our method with greedy and MBR decoding. More detailed comparative experiments are left for future work.

The utility function in MBR decoding In the MBR decoding compared in our experiments, we used edit-level metrics such as ERRANT or n -gram-level metrics such as GLEU as a utility function. This is based on the idea that it is appropriate to use the same metric as the evaluation measure for the utility function. Furthermore, since we used the $F_{0.5}$ score for ERRANT and JP-ERRANT, which emphasizes precision, we assumed that over-correction would be sufficiently reflected in the score. However, it is possible that a metric reflecting the degree of over-correction more accurately could be constructed, and the performance of MBR decoding might be underestimated. Despite such concerns, we believe that the experimental setup of this paper is appropriate. Additionally, we also tested cases where ERRANT was used as the utility function when GLEU was the evaluation metric, but no differences were observed.

Ethical Considerations

If many of the candidates generated by the LLM contain harmful outputs, the proposed method may fail to remove them and instead include them in the final output. However, since it is rare for harmful outputs to be generated as the exact same surface-

level text across different candidates, we believe they can generally be eliminated by setting the edit count threshold τ to a sufficiently high value. Empirical verification of this point is left for future work. At least in our manual analysis for the case study, we did not observe any harmful content.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work has been supported by JST SPRING. Grant Number JPMJSP2140.

References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.
- 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, 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.
- Bin Cao, Kai Jiang, Fayu Pan, Chenlei Bao, and Jing Fan. 2024. [Improving grammatical error correction by correction acceptability discrimination](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8818–8827, Torino, Italia. ELRA and ICCL.
- 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.

- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipour, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.
- Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024. [mbrs: A library for minimum Bayes risk decoding](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 351–362, Miami, Florida, USA. 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.
- Bryan Eikema and Wilker Aziz. 2020. [Is MAP decoding all you need? the inadequacy of the mode in neural machine translation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. [Grammatical error correction in low error density domains: A new benchmark and analyses](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.
- Vaibhava Goel and William J Byrne. 2000. [Minimum bayes-risk automatic speech recognition](#). *Computer Speech Language*, 14(2):115–135.
- 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 Goto, Yusuke Sakai, and Taro Watanabe. 2025a. [gec-metrics: A unified library for grammatical error correction evaluation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. [Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human?](#) In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- 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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Masahiro Kaneko and Naoaki Okazaki. 2023. [Reducing sequence length by predicting edit spans with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029, Singapore. Association for Computational Linguistics.
- Anisia Katinskaia and Roman Yangarber. 2024. [GPT-3.5 for grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Revisiting meta-evaluation for grammatical error correction](#). *Transactions of the Association for Computational Linguistics*, 12:837–855.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023a. [Efficient memory management for large language model](#)

- serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023b. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. 2025. Edit-wise preference optimization for grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3401–3414, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ruixi Lin and Hwee Tou Ng. 2021. System combination for grammatical error correction based on integer programming. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 824–829, Held Online. INCOMA Ltd.
- 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.
- 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.
- 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. Preprint, arXiv:1605.02592.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashkyi. 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.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhashkyi, Artem Chernodub, Oleksandr Kornienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Junyeul Park. 2025. Multilingual grammatical error annotation: Combining language-agnostic framework with language-specific flexibility. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 202–212, Vienna, Austria. Association for Computational Linguistics.
- 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](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Vyas Raina and Mark Gales. 2023. [Minimum Bayes’ risk decoding for system combination of grammatical error correction systems](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 105–112, Nusa Dua, Bali. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.
- Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe. 2024. [Toward the evaluation of large language models considering score variance across instruction templates](#). In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 499–529, Miami, Florida, US. Association for Computational Linguistics.
- Ujjwal Sharma and Pushpak Bhattacharyya. 2025. [Hi-GEC: Hindi grammar error correction in low resource scenario](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6063–6075, Abu Dhabi, UAE. Association for Computational Linguistics.
- Alexey Sorokin. 2022. [Improved grammatical error correction by ranking elementary edits](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. 2025. [Adapting LLMs for minimal-edit grammatical error correction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 118–128, Vienna, Austria. Association for Computational Linguistics.
- Oleksiy Syvokon and Mariana Romanyshyn. 2023. [The UNLP 2023 shared task on grammatical error correction for Ukrainian](#). In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 132–137, Dubrovnik, Croatia. Association for Computational Linguistics.
- Chenming Tang, Fanyi Qu, and Yunfang Wu. 2024. [Ungrammatical-syntax-based in-context example selection for grammatical error correction](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1758–1770, Mexico City, Mexico. Association for Computational Linguistics.
- Maksym Tarnavskiy, 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.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). Preprint, arXiv:2503.19786.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. [Gemma 2: Improving open language models at a practical size](#). Preprint, arXiv:2408.00118.
- Junrui Wang, Mengyang Qiu, Yang Gu, Zihao Huang, and Jungyeul Park. 2025. [Refined evaluation for end-to-end grammatical error correction using an alignment-based approach](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 774–785, Abu Dhabi, UAE. 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Helen Yannakoudakis, Øistein E Andersen, Ardeshtir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. [Developing an automated writing placement system for ESL learners](#). *Applied Measurement in Education*, 31(3):251–267.
- Soyoung Yoon, Sungjoon Park, Gyuwan Kim, Junhee Cho, Kihyo Park, Gyu Tae Kim, Minjoon Seo, and Alice Oh. 2023. [Towards standardizing Korean grammatical error correction: Datasets and annotation](#). In

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6713–6742, Toronto, Canada. Association for Computational Linguistics.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. **SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022. **SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. **Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Yike Zhao, Xiaoman Wang, Yunshi Lan, and Weining Qian. 2025. **UnifiedGEC: Integrating grammatical error correction approaches for multi-languages with a unified framework**. In *Proceedings of the 31st International Conference on Computational Linguistics: System Demonstrations*, pages 37–45, Abu Dhabi, UAE. Association for Computational Linguistics.

A Actual template

Figure 4 shows the template for English experiments, and Figure 5 for other language experiments. The symbols of [SOURCE] and [FEWSHOT] will be replaced with actual input sentence and few-shot examples during inference.

B Detailed Experimental Settings

For the experiments in Table 2, we conducted reproduction experiments for GECToR and T5, respectively.

GECToR. GECToR was trained in three stages following Omelanchuk et al. (2020). Regarding the datasets, the first stage used PIE-synthetic (Awasthi et al., 2019). The second stage used only pairs from the BEA-2019 (Bryant et al., 2019) training data that contained at least one correction, and the third

You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling in the input sentence. Make the smallest possible change in order to make the sentence grammatically correct. Change as few words as possible. Do not rephrase parts of the sentence that are already grammatical. Do not change the meaning of the sentence by adding or removing information. If the sentence is already grammatically correct, you should output the original sentence without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:

Figure 4: Instruction template for **English** datasets. [FEWSHOT] and [SOURCE] will be replaced with actual few-shot examples and an input sentence, respectively.

You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling of the input essay written by a learner of [LANG]. Make the smallest possible change in order to make the essay grammatically correct. Change as few words as possible. Do not rephrase parts of the essay that are already grammatical. Do not change the meaning of the essay by adding or removing information. If the essay is already grammatically correct, you should output the original essay without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:

Figure 5: Instruction template for datasets of **other than English**. [LANG] will be replaced with a language name, such as “Czech” or “German.”

stage used the entire W&I-LOCNESS (Yan-nakoudakis et al., 2018) training data. We used bert-base-cased⁶ (Devlin et al., 2019) and deberta-v3-large⁷ (He et al., 2023) as the models. The training settings also basically followed Omelanchuk et al. (2020), except that the first stage was set to 10 epochs. During inference, the keep confidence and minimum error probability thresholds were determined for each dataset. We tried both values from {0.1, 0.2, . . . , 1.0} and used the values that maximized the metrics on the development data for inference on the evaluation data. Our reproduction results are competitive with the values reported by Omelanchuk et al.

⁶<https://huggingface.co/google-bert/bert-base-cased>

⁷<https://huggingface.co/microsoft/deberta-v3-large>

(2020) and Tarnavskyi et al. (2022).

T5. Following Rothe et al. (2021), T5 was fine-tuned using 2,372,119 sentences from the cLang8 dataset. We used t5-v1_1-large⁸ as the base model. The model was trained for 10 epochs with a learning rate of 1e-5. Our reproduction achieves the score of $F_{0.5} = 73.4$ on the BEA-2019 test set as shown in Table 2, which exceeds the T5-large score of $F_{0.5} = 72.06$ reported by Rothe et al. (2021).

C Actual Templates

Table 6 shows 10 templates used in Section 6.3.

ID	Template
1	You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling in the input sentence. Make the smallest possible change in order to make the sentence grammatically correct. Change as few words as possible. Do not rephrase parts of the sentence that are already grammatical. Do not change the meaning of the sentence by adding or removing information. If the sentence is already grammatically correct, you should output the original sentence without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
2	Make minimal changes to the following text such that it is grammatically correct. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
3	Please correct the following text. Do not attempt to rewrite it into perfect English or to interpret the text. Often, things could be expressed better by paraphrase, but the task is to make minimal changes to correct the text. Do not change anything that is correct. Please make no changes if there are no errors. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
4	Reply with a corrected version of the input sentence with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
5	Correct the grammatical errors in the following sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
6	Revise mistakes in this text. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
7	Rewrite the following text with proper grammar. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
8	Improve the grammar of this text. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
9	Correct the following to standard English. Return only the corrected text and nothing more.\n[FEWSHOT]\nSentence: [SOURCE]\nCorrection:
10	Fix the errors in this sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:

Table 6: The ten templates used in the experiments of this paper. [SOURCE] and [FEWSHOT] are replaced by the erroneous sentence and few-shot examples, respectively.

⁸https://huggingface.co/google/t5-v1_1-large

Zero-Shot Phonics: Evaluating Constraint-Adherent Phonics Story Generation in Large Language Models

Maria Monica Manlises

De La Salle University

Manila, Philippines

maria_monica_manlises@dlsu.edu.ph

Ethel Ong

De La Salle University

Manila, Philippines

ethel.ong@dlsu.edu.ph

Abstract

Phonics stories are essential for early literacy, requiring controlled repetition of grapheme-phoneme (GP) patterns while remaining simple, readable, and developmentally appropriate. Generating such texts poses a challenge for large language models (LLMs), which must handle multiple linguistic and pedagogical constraints. To investigate how these constraints can be balanced, we systematically vary prompt design across 16 configurations and evaluate six LLMs in a zero-shot setting, producing 8,688 outputs (39,096 stories). Outputs are assessed using a multi-dimensional framework covering phonological alignment, developmental lexical appropriateness, readability, and narrative quality. Results show that while LLMs generate highly readable and age-appropriate text, they exhibit variability in phoneme control and narrative coherence. Prompt design and model choice significantly affect performance, revealing trade-offs across constraints. These findings highlight the challenges of controllable educational text generation and the importance of prompt design in balancing instructional objectives. We release our prompts, generated stories, and evaluation framework to support future work in phonics-based story generation.

1 Introduction

Phonics stories are a foundational tool in early literacy instruction, designed to reinforce grapheme-phoneme (GP) correspondences through controlled and repetitive exposure. However, unlike general narrative text, these stories must satisfy multiple instructional constraints, requiring consistent reinforcement of target sound patterns while remaining simple, readable, and developmentally appropriate for early learners (National Reading Panel, 2000; Ehri, 2020).

Phonics instruction plays a critical role in early literacy and speech development, teaching learners to decode written language by associating

graphemes—letters or letter combinations—with their corresponding phonemes—sounds or sound combinations (National Reading Panel, 2000; Ehri, 2020). Decodable texts, including phonics stories, are essential components of systematic phonics instruction, as they support the development of phonetic decoding skills through structured GP reinforcement (Ehri, 2020). For example, a phonics story targeting the grapheme ‘s’ and the phoneme /s/ may include sentences such as “*Sam sees the snake in the grass.*” By reinforcing GP decoding, these stories serve as effective tools for speech and reading development (Cheatham and Allor, 2012; Pennell et al., 2024; Murphy Odo, 2024).

Large language models (LLMs) have demonstrated strong capabilities in generating coherent and contextually appropriate text, including narratives, with minimal prompting (Xie et al., 2023; Tian et al., 2024). Prior work has explored guiding LLMs to produce age-appropriate children’s stories through prompt design and lexical simplification (Valentini et al., 2023). However, existing work does not evaluate whether LLMs can simultaneously satisfy phonological and pedagogical constraints. Unlike general storytelling, this task requires balancing readability and developmental appropriateness with consistent and meaningful reinforcement of target GP mappings.

In this paper, we investigate the effectiveness of zero-shot prompting in generating phonics stories that emphasize specific GP mappings. We evaluate outputs using a multi-dimensional framework spanning phonological alignment, developmental lexical appropriateness, readability, and narrative quality.

To clarify terminology used throughout this work, while the terms “decodable texts” and “phonics stories” are often used interchangeably, we use “phonics stories” to refer specifically to short, structured narratives designed for early readers, whereas “decodable texts” more broadly include instruc-

tional materials that emphasize controlled GP exposure without necessarily maintaining narrative structure. In this work, we primarily use the term “phonics stories”, while also examining the effect of terminology in prompt design.

Our main contributions are as follows:

1. We use zero-shot prompting for large-scale phonics story generation across 16 prompt configurations and 6 LLMs, producing a dataset¹ of 8,688 outputs (39,096 stories).
2. We define a multi-dimensional evaluation framework comprising 12 metrics that capture phonological alignment, developmental lexical appropriateness, readability, and narrative quality.
3. We provide a detailed analysis of prompt design and model choice, identifying trade-offs between phonological accuracy, linguistic simplicity, and narrative coherence.

2 Related Work

2.1 Automated Children’s Story Generation

Automated story generation evolved from early rule-based systems such as TALE-SPIN (Meehan, 1977) and MINSTREL (Turner, 1992) to modern neural approaches (Roemmele et al., 2017). While early systems lacked flexibility and relied heavily on handcrafted rules, subsequent data-driven methods improved narrative coherence through structured knowledge integration and learned representations (Liu and Singh, 2002; Roemmele et al., 2017).

More recently, the emergence of LLMs has significantly advanced the quality of generated stories, achieving improvements in fluency, coherence, and contextual relevance (Brown et al., 2020). However, despite these advances, instructionally constrained storytelling, particularly for educational purposes such as phonics story generation, remains underexplored, as most work prioritizes general narrative quality over pedagogical alignment.

2.2 LLM Capabilities

2.2.1 Story Generation

Recent studies consistently show that LLMs outperform earlier neural models in story generation tasks, particularly in terms of fluency and overall narrative quality (Xie et al., 2023). However, these

gains are accompanied by limitations in originality and control, especially when compared to human-authored stories.

In the context of children’s storytelling, Valentini et al. (2023) further demonstrate that while LLMs can generate readable and coherent narratives, they often struggle to satisfy developmental constraints such as age-appropriate vocabulary in zero-shot settings. Their findings highlight the importance of lexical simplification, which significantly improves both readability and alignment with target age groups. Complementing this, Alasmari et al. (2025) move beyond surface-level fluency by proposing an automated framework for evaluating narrative cohesion grounded in Gérard Genette’s narrative theory. By operationalizing cohesion into computable features and achieving high classification performance, their work provides a foundation for systematically evaluating and improving story structure in both human-authored and AI-generated texts.

Together, these studies suggest that while LLMs are strong generators of fluent narratives, ensuring developmental appropriateness and structural quality requires additional mechanisms for control and evaluation.

2.2.2 Constrained Generation

Following explicit constraints specified in prompts is not just a challenge in general children’s storytelling but is also central in phonics-based story generation of LLMs. Prior work shows that even relatively simple constraints, such as age-level requirements, are not consistently satisfied in zero-shot settings (Valentini et al., 2023), raising concerns about the reliability of instruction-following in educational contexts.

More broadly LLMs’ controlled generation capabilities have been evaluated across various constraints such as sentiment conditioning, keyword inclusion, prefix constraints, and numerical planning where adherence was found to vary depending on task formulation (Sun et al., 2023). Similarly, Marbun and Shishido (2026) demonstrated how readability targets in prompt formulation can improve lexical appropriateness compared to traditional prompting strategies.

Collectively, these works emphasize that while LLMs possess strong generative capabilities, achieving reliable constraint adherence requires deliberate prompt design and task-specific strategies.

¹<https://huggingface.co/datasets/monicamanlises/zero-shot-phonics>

2.2.3 Phonological Tasks

In order to generate phonics stories, LLMs must exhibit some degree of phonological awareness. While LLMs demonstrate strong theoretical knowledge of phonetics and phonology in controlled evaluations, they often struggle to apply this knowledge in generative settings (Peng et al., 2025), indicating a gap between passive understanding and practical use. This limitation is further supported by Suvarna et al. (2024), who evaluate LLMs on tasks such as grapheme-to-phoneme (G2P) conversion, syllable counting, and rhyme generation. Although advanced models perform relatively well among LLMs, they still fall short of rule-based systems and human performance, particularly in tasks requiring fine-grained phonological precision.

These findings highlight a key challenge for phonics story generation: while LLMs can approximate phonological reasoning, their limitations necessitate careful model selection and evaluation.

3 Task Definition

We define phonics story generation as a constrained phonological text generation task. Given a target GP pair or phonic set, the objective is to generate a short story that simultaneously satisfies three main requirements:

1. **Phonological Alignment:** The story should reinforce the target GP mapping through repeated and meaningful usage.
2. **Developmental Appropriateness:** The vocabulary, sentence structure, and overall readability should align with the intended reading level of early learners, ensuring that texts are both accessible and instructionally effective.
3. **Narrative Quality:** The story should be coherent, cohesive, and easy to follow.

Unlike general story generation, this task requires balancing competing constraints. Increasing phoneme repetition may improve phonological reinforcement but reduce narrative naturalness, while prioritizing readability may weaken phonological control. As a result, phonics story generation serves as a challenging testbed for controllable text generation under educational constraints.

4 Method

This study follows a generation-evaluation pipeline to assess the capability of LLMs in generating

phonics stories for early readers. In a zero-shot setting, multiple LLMs were prompted to produce short phonics stories and decodable texts designed to reinforce specific GP correspondences. The generated outputs were subsequently evaluated using a multi-dimensional framework encompassing phonological target alignment, developmental appropriateness, readability, and narrative quality. This methodology enables a systematic investigation of LLM performance in constrained educational text generation.

4.1 Models

We evaluated six LLMs: Claude-3.7-Sonnet (claude), GPT-4o (gpt4o), GPT-4o-Mini (gpt4o_mini), Gemini-2.0 Flash (gemini), DeepSeek-V3 (deepseek), Mistral-Large-2.1 (mistral). These model versions were available as of March 2025.

The models claude, gpt4o, and gpt4o_mini were selected due to their demonstrated strengths in phonological subtasks such as grapheme-to-phoneme (G2P) conversion and syllable counting in prior benchmarks (e.g., PhonologyBench (Suvarna et al., 2024)). Batched API calls were used for these models.

To have a balance of three open-source and three closed-source models, mistral, deepseek, and gemini were also included. These were accessed via standard single-prompt endpoints.

4.2 Prompt Design

We designed 16 structured prompts grouped into three prompt sets. These were constructed to evaluate the ability of LLMs to adhere to phonological, developmental, and narrative constraints. The prompts targeted either individual GP pairs or phonic sets and varied in their degree of constraint and instructional framing.

4.2.1 Phoneme Selection

Target GP mappings are derived from Pearson’s *Phonics Progression Chart* (Pearson Education, n.d.), which follows a structured synthetic phonics curriculum aligned with established literacy research (Ehri, 2020; National Reading Panel, 2000; Rose, 2006). Graphemes are preserved in orthographic form, while phonemes are represented in ARPABET. Table 3 lists these GP pairs, as well as corresponding target age, phonic phase, and phonic set.

4.2.2 Prompt Variants

All prompts were initialized with a shared system instruction (Figure 1) that defines the model’s role and objective. Three prompt sets were then constructed, each with a distinct focus, along with minor variations within each set.

To examine consistency, half of the prompts required the model to generate a single story, while the remaining prompts required the generation of eight stories. Additionally, prompts alternated between the terms “phonics stories” and “decodable texts” to analyze the effect of terminology on generation behavior, despite their conceptual overlap.

The full set of prompt variants is provided in Appendix A.

Prompt Set 1: Single Grapheme-Phoneme Pair and Narrative Focused Generation (Prompts 1-8) This set targeted individual GP pairs. Variants included basic prompts and plot-driven prompts. These prompts evaluated the ability of LLMs to reinforce a single phoneme within a coherent narrative.

Prompt Set 2: Phonic Set Focused Generation (Prompts 9-12) This prompt set introduced constraints based on a given phonic set, emphasizing the GP pairs within the set while permitting supporting GP pairs from earlier sets to improve fluency and phonic progression.

Prompt Set 3: Age-Constrained Generation (Prompts 13-16) This set incorporated developmental constraints. Phases 2-3 were designed for children aged 4–5, while Phase 5 was designed for children aged 5–6. These age ranges were originally aligned with the UK Department for Education’s *Letters and Sounds* framework, in which Phases 2 and 3 are introduced during Reception (ages 4–5) and Phase 5 is taught in Year 1 (ages 5–6) (Department for Education and Skills, 2008). Although *Letters and Sounds* was formally withdrawn in 2024 as part of the shift toward updated validated systematic synthetic phonics programs, its phase structure remains widely referenced in both research and instructional materials. This progression is also reflected in Pearson’s Phonics Progression Chart and broader systematic synthetic phonics curricula (Pearson Education, n.d.; Rose, 2006).

4.3 Data Cleaning and Post-processing

Generated outputs were standardized for consistency and evaluability. The process included removing extraneous labels, re-prompting malformed outputs, and normalizing formatting.

While the target structure specifies exactly four sentences, minor deviations (3-5 sentences) were retained to avoid introducing bias through aggressive filtering. These cases were included in evaluation to preserve the integrity of model outputs.

4.4 Evaluation Metrics

Generated stories are evaluated using 12 automated metrics spanning four dimensions: phonological alignment, developmental lexical appropriateness, readability, and narrative quality. Full formal definitions are provided in Appendix B. For clarity, metrics are referenced by both name and index (1-12) throughout the paper.

4.4.1 Phonological Target Alignment (Metrics 1-4)

Phonological target alignment measures adherence to target GP mappings. A word is considered aligned if the target grapheme appears in the word and the corresponding phoneme appears in its ARPABET transcription. Metrics include (M1) **Story-level Phoneme Density**, (M2) **Sentence-level Phoneme Density**, (M3) **Mean Phonics Level Deviation**, and (M4) **Maximum Phonics Level Deviation**.

4.4.2 Developmental Lexical Appropriateness (Metrics 5-6)

Lexical appropriateness captures the suitability of vocabulary for early readers. This is quantified using Age of Acquisition (AoA) norms (Kuperman et al., 2012; Valentini et al., 2023). Metrics include (M5) **Mean AoA Deviation** and (M6) **Maximum AoA Deviation** from the target age range.

4.4.3 Readability (Metrics 7-10)

Readability measures the linguistic simplicity of the text and is assessed using standard metrics, including (M7) **Flesch Reading Ease**, (M8) **Flesch-Kincaid Grade Level**, (M9) **Gunning Fog Index**, and (M10) **Automated Readability Index**. Thresholds aligned with early readers (ages 4-6) are used for interpretation (Table 4).

4.4.4 Narrative Quality (Metrics 11-12)

Narrative quality evaluates coherence and cohesion in the generated stories. These metrics

are scored using an LLM-as-a-judge framework (Claude-Haiku-4.5) with rubric-based prompting to improve consistency and reduce variance. Scores are normalized to a 0-1 scale. **(M11) Coherence** measures global narrative organization, including setup, logical progression, causal relationships, and resolution. **(M12) Cohesion** assesses local connectedness across sentences, including referential clarity, lexical consistency, and the absence of contradictions.

The full evaluation prompt, inspired by the coherence and cohesion frameworks of Alasmari et al. (2025) and Hargood et al. (2011), is provided in Appendix G.

The evaluator models also generated textual justifications alongside their rubric-based scores. These outputs were retained separately for qualitative inspection but were omitted from the main dataset and analyses to maintain focus on aggregate quantitative comparisons.

4.5 Statistical Analysis

Evaluation results were compared across metric, prompt design, and model levels using aggregate analysis, omnibus tests, and post-hoc pairwise comparisons.

Prior to significance testing, a Shapiro-Wilk normality test was applied to each group within a comparison. The test evaluates the null hypothesis that a sample is drawn from a normal distribution; a p -value ≤ 0.05 indicates a significant deviation from normality.

As most metric distributions violated normality assumptions, non-parametric tests were used throughout. For comparisons involving more than two groups, the Kruskal-Wallis (KW) test was applied as the omnibus test. For pairwise comparisons between two groups, the Mann-Whitney U (MWU) test was used.

For comparisons with more than two groups, post-hoc pairwise tests were conducted only when the omnibus test was significant ($\alpha = 0.05$). Bonferroni correction was applied to adjust p -values and control the family-wise error rate.

Omnibus tests determine whether significant differences exist across groups, while post-hoc pairwise tests identify which specific group pairs differ significantly.

Metric	Mean	Std
1. Story-Level Phoneme Density ($\uparrow$)	0.360	0.222
2. Sentence-Level Phoneme Density ($\uparrow$)	0.373	0.219
3. Mean Phonics Level Deviation ($\downarrow$)	1.440	0.997
4. Max Phonics Level Deviation ($\downarrow$)	1.053	0.728
5. Mean AoA Deviation ($\downarrow$)	0.206	0.275
6. Max AoA Deviation ($\downarrow$)	3.378	1.637
7. Flesch Reading Ease ($\uparrow$)	104.206	11.721
8. Flesch-Kincaid Grade Level ($\downarrow$)	0.196	1.681
9. Gunning Fog Index ($\downarrow$)	2.848	1.217
10. Automated Readability Index ($\downarrow$)	-0.386	2.320
11. Coherence ($\uparrow$)	0.665	0.231
12. Cohesion ($\uparrow$)	0.704	0.167

Table 1: Summary statistics across all generated stories. Arrows indicate whether higher ($\uparrow$) or lower ($\downarrow$) values are generally preferable.

5 Results and Discussion

Through zero-shot prompting, a dataset of 8,688 outputs comprising 39,096 stories was generated, cleaned, and evaluated across four metric groups. Overall, models consistently produce highly readable and age-appropriate text, but show greater variability in phonological alignment and narrative quality. Both prompt design and model choice significantly affect performance across all metrics.

5.1 Metric-Level Comparison

Table 1 summarizes aggregate performance across all models and prompt configurations. Results indicate consistently high readability and appropriate vocabulary usage, alongside moderate adherence to phonological constraints and more variable narrative quality (see Appendix F).

5.1.1 Phonological Target Alignment

Phoneme density is moderate but acceptable (0.360), indicating emphasis of target phonemes in at least one per three words. Sentence-level density (0.373) is slightly higher, suggesting uneven distribution across sentences. However, phonics level deviation further highlights weak control. The mean deviation (1.44) indicates frequent use of words outside the target level, while a high maximum deviation reflects occasional inclusion of a substantially different phonic phase. The raw values show that around 75% of deviation is negative. Along with a lower maximum phonics level deviation, this indicates the use of phonemes in easier phonic phases.

5.1.2 Developmental Lexical Appropriateness

Lexical appropriateness is generally strong, with low mean AoA deviation (0.206). However, a high

average maximum deviation (3.378) indicates occasional out-of-scope words, even reaching a deviation of 10.920. This reflects stable average performance but inconsistent worst-case behavior, suggesting limited constraint enforcement at the token level.

5.1.3 Readability

Readability is consistently high, with Flesch Reading Ease above 100 and grade-level metrics near or below early elementary levels. This indicates strong control over linguistic simplicity, possibly due to the maximum syllable requirement in the basic prompt in Figure 1.

Compared to other metrics, readability aligns closely with general language modeling objectives, making it the most stable dimension. However, this strength does not guarantee alignment with phonics progression, as linguistically simple text may still fail to systematically reinforce target grapheme-phoneme correspondences.

5.1.4 Narrative Quality

Narrative quality is moderate, with coherence (0.665) lower than cohesion (0.704). This indicates stronger local sentence connections than global narrative structure. Performance varies across models, with stronger models producing more stable narrative flow and weaker models showing greater inconsistency.

5.2 Prompt Design-Level Comparison

Prompt design significantly affects performance across all metric groups, influencing trade-offs between phonological alignment, readability, and narrative quality. Different aspects of prompt design, such as prompt set, terminology, and story number, were evaluated for their performance. Figures 5 and 6 summarize performance across prompt sets and prompts, respectively.

5.2.1 Comparison Across Prompt Sets

All prompt sets achieve high readability ratings, with Prompt Set 2 producing the easiest texts.

Prompt Set 1, which was designed to focus on both specific phonemes and plot, achieves desired outputs. It yields the strongest phonemic alignment but with at least 1 level of deviation from the desired phonic phase. It also achieves a moderate narrative quality, only a close second to Prompt Set 3. With regards to lexical appropriateness, it achieves the lowest mean deviation but also contains more

words that deviate greatly from the target age of acquisition.

Meanwhile, Prompt Set 2, which focuses on phonic-set-based generation, has the largest deviation from the desired phonics phase, which may be because it is more lenient on what specific phoneme to use. The texts produced by this prompt set also had the worst coherence and cohesion.

Prompt Set 3, which focuses on age-appropriateness, produces texts with more consistent ages of acquisition rates and produces acceptable phonological alignment, although it is the lowest in terms of phoneme density.

In Figure 10, KW tests revealed that there were significant differences between the prompt sets for all metrics except for metric 3 (Maximum Phonics Level Deviation). Bonferroni-corrected MWU pairwise tests show that all prompt sets are significantly different from each other with regard to readability despite producing extremely readable texts.

Similarly, there are significant differences across all specific prompts on all metrics except for phonics level deviation. Narrative coherence had the most pairs with significant differences (91 out of $\binom{16}{2} = 120$).

5.2.2 "Phonics-Based Story" vs "Decodable Text"

The prompts alternated between using "phonics-based stories" and "decodable texts." No significant differences are observed in phoneme density, indicating stable phonological alignment across the terminology used. However, using "phonics-based story" achieves higher coherence and cohesion, while "decodable text" prompts produce easier texts. This difference was found to be significant using MWU tests (Figure 10).

This suggests that LLMs may understand that stories require richer narratives, while texts simply adhere to the target of being decodable. Thus, even a slight change in the terminology for the target output influences the trade-off between narrative structure and readability, although the effect is less pronounced than other design factors.

5.2.3 Single-story vs Multi-story Prompts

Using MWU tests, we compared each single-story prompt to its equivalent multi-story prompt (e.g., Prompt 1 vs.3), totaling 8 pairs. This showed that all single-story prompts achieve significantly higher phoneme density, while multi-story prompts yield better coherence, cohesion, and readability.

This trade-off is consistent across prompt sets, indicating that output format strongly affects constraint adherence. There was no significant difference found for phonics level deviation metrics.

5.3 Model-Level Comparison

Model choice is the primary determinant of overall performance, showing significant differences across all metrics using KW tests and post-hoc Bonferroni-corrected MWU tests. Particularly, performance differences in terms of phoneme density are most evident and are present in all $\binom{6}{2} = 15$ pairs. Further analysis showed that some models showed clear trade-offs with phonological alignment for narrative quality while others show moderate yet acceptable performance across all metrics.

Figure 8 shows claude and gemini have the strongest phonological alignment, where the target phoneme or phonic set is present in around one per two words. However, they also perform the worst in terms of narrative quality. In contrast, gpt4o_mini and mistral rank highest in terms of narrative quality but lowest in terms of phonological alignment. Keeping a better balance of both these metric categories are deepseek and gpt4o, with deepseek showing the best set of balanced scores as seen in Figure 7

In samples 1 to 6 in Table 6, we can see stories generated by the models for the GP mapping between grapheme ‘le’ and phonemes [‘L’, ‘AH L’]. This is one case that highlights the trade-off between phonological alignment and narrative quality, where we see that the stories generated by claude, gemini, and deepseek have the target GP pair appearing in more than 1 per 2 words while having low narrative quality. In contrast, mistral and gpt4o_mini gained perfect scores for coherence and cohesion, but the target GP pair appeared in fewer than 1 in 3 words in the story.

5.3.1 Selected Model Performance Comparison

Figure 9 presents a focused comparison between claude—because of its strong alignment—and deepseek—because of its strong balanced performance. They are compared across four representative metric categories: phonological alignment (M1), lexical difficulty control (M5), readability (M7), and narrative quality (M11-M12), evaluated across all prompt sets (PS1-PS3).

Phonological Alignment (M1) Across all prompt sets, claude consistently achieves higher

phoneme coverage, with median values centered around 0.45–0.55, compared to deepseek’s lower range of approximately 0.30–0.45. This confirms its stronger adherence to target GP constraints. However, claude also exhibits slightly wider interquartile ranges, indicating less consistent enforcement. While weaker in overall coverage, deepseek, shows more stable distributions across prompt configurations.

Lexical Difficulty Control (M5) Both models maintain relatively low AoA deviation, suggesting that generated vocabulary generally aligns with the intended difficulty level. Stories by claude show tighter distributions closer to zero, indicating more consistent lexical control, while deepseek, particularly under PS2, exhibits slightly higher medians and greater spread, suggesting occasional introduction of more difficult words.

Readability (M7) In terms of readability, deepseek clearly outperforms claude. Across all prompt sets, deepseek achieves higher Flesch Reading Ease scores, with medians consistently exceeding those of claude. Furthermore, deepseek’s distributions are more compact, indicating more stable control over sentence complexity. More variability is seen in claude’s stories, showing more low-end outliers and reflecting occasional drops in readability.

Narrative Quality (M11–M12) Stories generated by deepseek demonstrate stronger narrative performance overall, with higher median scores in both coherence and cohesion across all prompt sets. The gap is especially pronounced under PS2, where claude exhibits a notable decline and increased variance. While claude improves under PS3, it remains slightly below deepseek in both metrics. Tighter distributions present in deepseek’s stories indicate more consistent narrative structure.

Trade-off Analysis These results highlight a clear trade-off between constraint satisfaction and overall story quality. While claude excels in enforcing phonological constraints and maintaining lexical simplicity, it comes at the cost of reduced readability and weaker narrative structure. In contrast, deepseek provides a more balanced performance, achieving higher readability and narrative quality while maintaining moderate phonological alignment. Sample stories 7 to 10 in Table 6 also reflect these trade-offs.

Prompt Effects Across both models, PS2 consistently underperforms relative to PS1 and PS3. This is reflected in lower phoneme coverage, higher lexical deviation, and reduced narrative quality. In contrast, PS1 and PS3 produce more stable and higher-quality outputs, with PS3 often achieving the best balance between constraint adherence and fluency.

6 Conclusion

This work investigates zero-shot phonics story generation as a constrained text generation task requiring simultaneous adherence to phonological alignment, developmental appropriateness, readability, and narrative quality. Through a large-scale evaluation across multiple models and prompt configurations, our findings reveal a consistent pattern: while LLMs produce highly readable and age-appropriate text, they exhibit moderate yet acceptable phonological alignment and variable performance in narrative coherence.

Our analysis highlights systematic trade-offs between constraint categories. Models that achieve stronger phonological alignment often do so at the expense of narrative quality and vice versa. Prompt design further influences these trade-offs, shaping how constraints are prioritized, while model choice determines the overall balance and stability of performance across metrics.

Importantly, results show that no single model consistently satisfies all instructional requirements, underscoring the difficulty of multi-constraint generation in educational contexts. While some models demonstrate more balanced behavior, phonological alignment remains the most challenging dimension, particularly in maintaining consistent and instructionally meaningful reinforcement of target grapheme-phoneme correspondences.

Overall, this work establishes phonics story generation as a stringent testbed for controllable text generation and provides a multi-dimensional evaluation framework for assessing instructional alignment. These findings highlight the need for methods beyond zero-shot prompting, including constrained decoding, structured generation, or fine-tuning approaches, to better align LLM outputs with pedagogical objectives.

By systematically analyzing model behavior across constraints, this study contributes both empirical insights and evaluation tools for future research in educational NLP and controllable text

generation.

Limitations

This study has several limitations that suggest directions for future work. First, the evaluated models reflect capabilities available as of early 2025; newer systems may exhibit improved performance. Consequently, greater emphasis should be placed on the generalizability of the proposed generation, evaluation, and analysis framework.

Story generation used default sampling parameters rather than deterministic decoding (e.g., temperature = 0), reflecting real-world usage but limiting reproducibility. In addition, all stories were constrained to four sentences, restricting the evaluation of longer narrative structures and potentially underestimating model capabilities in maintaining coherence over extended contexts.

Phonological alignment was measured using GP matching via G2P conversion. While scalable, this approach does not capture pronunciation variability or contextual phoneme realization. Readability was assessed using formula-based metrics that capture surface-level linguistic complexity but may not fully reflect comprehension difficulty for early learners. Narrative quality was evaluated using a single LLM-based evaluator. While rubric-based prompting improves consistency, reliance on one evaluator introduces potential bias. Incorporating multiple LLM judges, ensemble methods, or human annotation would improve robustness. Although evaluator-generated textual rationales and subcriterion-level assessments were retained during scoring, these qualitative traces were not systematically analyzed in the present work. Future research could examine these explanations to better understand evaluator behavior, improve interpretability, and identify recurring narrative strengths or failure modes across generated stories.

Beyond automated evaluation, the study does not include comparisons against professionally developed or expert-authored phonics stories. As a result, while the proposed metrics enable relative comparison across prompts and models, they do not yet establish normative or pedagogically optimal ranges for high-quality instructional materials. Future work could benchmark generated stories against curated human-authored phonics corpora to better contextualize metric distributions, such as phoneme density and readability, and to identify target ranges associated with effective instructional

design.

The study also does not include expert evaluation or in-field testing and therefore does not measure actual learning outcomes such as phoneme acquisition, reading fluency, or comprehension. Future work should incorporate educator assessment and classroom-based studies to evaluate pedagogical effectiveness.

Finally, this study is limited to English phonics story generation. Extending the framework to other languages and literacy contexts remains an important direction for future research.

Ethical Considerations

This work explores the use of LLMs for generating phonics stories intended for early literacy instruction. While such systems have the potential to support scalable educational content creation, they also raise several ethical considerations.

The generated phonics stories are designed to support early readers but they are not a substitute for professionally developed instructional materials. LLM-generated content may contain inaccuracies in phonological alignment, inappropriate vocabulary, or inconsistencies in narrative structure. They should therefore be used with educator oversight.

This study relies on automated evaluation metrics, including LLM-based assessment for narrative quality. While this enables large-scale analysis, it does not fully capture pedagogical effectiveness. The absence of expert and in-field evaluation means that conclusions about instructional quality should be interpreted with caution.

LLMs are trained on large-scale web data and may reflect biases present in their training corpora. Although phonics stories are typically simple, generated content may still include cultural, social, or linguistic biases that are not appropriate for young learners. Future work should include bias analysis and content filtering to ensure inclusivity and appropriateness.

References

Jawharah Alasmari, Mohammed Alzyoudi, Masheal Alshehri, Rana Alshammari, and Reyouf Aldakan. 2025. [An automated predictive model for evaluating narrative cohesion in children's stories: a computational linguistic approach considering g rard genette's narrative structure theory](#). *International Journal of Adolescence and Youth*, 30(1):2500527.

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, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jennifer P. Cheatham and Jill H. Allor. 2012. [The influence of decodability in early reading text on reading achievement: a review of the evidence](#). *Reading and Writing*, 25(9):2223–2246.

Department for Education and Skills. 2008. [Letters and sounds: Principles and practice of high quality phonics](#). Technical report, UK Department for Education and Skills, Nottingham, UK.

Linnea C. Ehri. 2020. [The science of learning to read words: A case for systematic phonics instruction](#). *Reading Research Quarterly*, 55(S1):S45–S60.

Charlie Hargood, David Millard, and Mark Weal. 2011. [Measuring narrative cohesion: A five variables approach](#). In *Narrative and Hypertext Workshop at Hypertext 11 (06/06/11)*.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. [Age-of-acquisition ratings for 30,000 English words](#). *Behavior Research Methods*, 44(4):978–990.

Hugo Liu and Push Singh. 2002. [Makebelieve: using commonsense knowledge to generate stories](#). In *Eighteenth National Conference on Artificial Intelligence*, page 957–958, USA. American Association for Artificial Intelligence.

Ronald William Marbun and Makoto Shishido. 2026. [A readability-driven prompting framework for accurate grade-specific efl narrative creation](#). *International Journal of Advanced Computer Science and Applications*, 17(1).

James R. Meehan. 1977. Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pages 91–98, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dennis Murphy Odo. 2024. [The use of decodable texts in the teaching of reading in children without reading disabilities: a meta-analysis](#). *Literacy*, 58(3):267–277.

National Reading Panel. 2000. [Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction: Reports of the subgroups](#). Technical Report NIH Publication No. 00-4754, National Institute of Child Health and Human Development, Washington, DC.

Pearson Education. n.d. [Phonics progression chart](#). Accessed 2025.

Linkai Peng, Baorian Nuchged, and Yingming Gao. 2025. [Spoken language intelligence of large language models for language learning](#). *Preprint*, arXiv:2308.14536.

Ashley E. Pennell, Rebecca Lee Payne Jordan, Kindel Turner Nash, Kerry Elson, and Woodrow Trathen. 2024. [A healthy diet for beginning readers: Decodable texts as part of a comprehensive literacy program](#). *The Reading Teacher*, 77(5):673–684.

Melissa Roemmele, Andrew S. Gordon, and Reid Swanson. 2017. [Evaluating story generation systems using automated linguistic analyses](#).

Jim Rose. 2006. [Independent review of the teaching of early reading](#). Technical report, UK Department for Education.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. [PhonologyBench: Evaluating phonological skills of large language models](#). In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. [Are large language models capable of generating human-level narratives?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

S.R. Turner. 1992. *MINSTREL, a Computer Model of Creativity and Storytelling*. UCLA Computer Science Department.

Maria Valentini, Jennifer Weber, Jesus Salcido, Téa Wright, Eliana Colunga, and Katharina von der Wense. 2023. [On the automatic generation and simplification of children’s stories](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3588–3598, Singapore. Association for Computational Linguistics.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023. [The next chapter: A study of large language models in storytelling](#). In *Proceedings of the 16th International Natural Language Generation Conference*, pages 323–351, Prague, Czechia. Association for Computational Linguistics.

A Prompt Design

All prompt sets incorporated the basic prompt structure in Figure 1 and had respective additional requirements listed in Figures 2, 3, and 4.

System Prompt:

You are a children’s story writer and children’s educator specializing in speech and language acquisition. You write decodable texts and phonics stories, which are engaging stories for early readers that reinforce grapheme-phoneme associations through repeated exposure to target sounds.

Story Requirements:

- (Prompt-specific additional requirements)
- The story must contain exactly four sentences.
- Each sentence must contain fewer than eleven syllables.
- The story must reinforce the target phoneme(s).

Output Format:

****Title****

Sentence 1
Sentence 2
Sentence 3
Sentence 4

Figure 1: Base prompt structure for phonics story generation.

Prompt Set 1: Single GP and Narrative Focus

Prompt: Generate a short phonics-based story focusing on the grapheme ‘{grapheme}’, which corresponds to the phoneme(s) ‘{phoneme(s)}’.

Additional Requirements:

- Each sentence must include at least 2 words containing the grapheme ‘{grapheme}’.

Figure 2: Prompt Set 1: Single GP and Narrative Focus

Across all prompt sets, systematic variations, as seen in Table 2 were applied. The phrase “phonics story” was replaced with “decodable text” in selected prompts. Prompts requesting a single story were modified to generate “8 different” stories. Additional narrative constraints (e.g., requiring a plot

Prompt Set 2: Phonic Set Focus

Prompt: Generate a short phonics story that gives primary emphasis to the target graphemes in the list {grapheme_set}, which correspond to the phonemes {phoneme_set}, respectively. The majority of the key vocabulary should highlight these target grapheme-to-phoneme correspondences. You may also include graphemes from {lower_grapheme_set_list} to create complete and readable words, but they should play a supporting role.

Additional Requirement:

- The story should be designed to reinforce recognition and pronunciation of the target grapheme-to-phoneme correspondences.

Figure 3: Prompt Set 2: Phonic Set Focus

Prompt Set 3: Age-Constrained

Prompt: Generate a short phonics story for children aged {age} years old. The phonics story must focus on the grapheme '{grapheme}', which corresponds to the phoneme(s) '{phoneme(s)}'.

Additional Requirements:

- The story must be age-appropriate and engaging.
- Each sentence must include at least 2 words containing the grapheme '{grapheme}'.

Figure 4: Prompt Set 3: Age-Constrained

with consistent characters) were included in selected prompts.

B Evaluation Metrics

Notation.

- Each story $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ consists of sentences.
- Let $W_i = \{w_{i1}, w_{i2}, \dots, w_{iN_i}\}$ be the words in sentence s_i .
- Let $W = \bigcup_{i=1}^4 W_i$ be all words in the story, with total size $|W| = \sum_{i=1}^4 N_i$.
- $\mathbf{1}(\cdot)$ is the indicator function.
- $L(w)$ denotes the set of phonics phases associated with word w .

- Ph_t denotes the target phoneme.
- PS_t denotes the target phonics set.
- PP_t denotes the target phonics phase.
- $A(w)$ denotes the Age of Acquisition (AoA) of word w .
- $\mathcal{A} \subseteq W$ is the set of words with valid AoA values.
- $[a_{\min}, a_{\max}]$ is the target age range.

B.1 Metric 1: Phoneme Density (Story-Level)

Measures the proportion of words in the story that contain the target phoneme or a phoneme from the target set (for Prompt Set 2). This captures how strongly the story reinforces the target at the overall story level. Higher values indicate better phonological alignment.

$$M_1 = \frac{1}{|W|} \sum_{w \in W} \mathbf{1}(w \text{ contains phoneme}) \quad (1)$$

B.2 Metric 2: Phoneme Density (Sentence-Level)

Measures the average proportion of words containing the target phoneme or a phoneme from the target set (for Prompt Set 2) computed at the sentence level. This ensures that phoneme usage is distributed consistently across sentences rather than concentrated in only one part of the story.

$$M_2 = \frac{1}{4} \sum_{i=1}^4 \left(\frac{1}{|W_i|} \sum_{w \in W_i} \mathbf{1}(w \text{ contains phoneme}) \right) \quad (2)$$

B.3 Metric 3: Mean Phonics Level Deviation

Measures the deviation of the average phonics phases associated with each word from the target phonics phase. Note that tricky words listed in 3 were counted towards that phonics phase. Values closer to 0 indicate better alignment.

$$M_3 = \left| \frac{1}{|W|} \sum_{w \in W} \left(\frac{1}{|L(w)|} \sum_{\ell \in L(w)} \ell \right) - PP_t \right| \quad (3)$$

Variation	Prompt IDs
"phonics-based story" → "decodable text"	2, 4, 6, 8, 10, 12, 14, 16
Single story → "8 different" stories	3, 4, 7, 8, 11, 12, 15, 16
Plot requirement added	5, 6, 7, 8

Table 2: Systematic prompt variations applied across prompt sets.

Target Age	Phase	Set	Grapheme-Phoneme Mappings	Tricky Words
4-5	2	1	s→['S'], a→['AA','AE','AH'], t→['T'], p→['P']	the, to, I, no, go, into, and
		2	i→['IH'], n→['N'], m→['M'], d→['D']	
		3	g→['G'], o→['AA'], c→['K'], k→['K']	
		4	ck→['K'], e→['EH'], u→['AH'], r→['R']	
		5	h→['HH'], b→['B'], f→['F'], ff→['F'], l→['L'], ll→['L'], ss→['S']	
	3	6	j→['JH'], v→['V'], w→['W'], x→['K','S']	he, she, we, me be, was, you, they, all, are, my, her
		7	y→['Y'], z→['Z'], zz→['Z'], qu→['K','W']	
		8	ch→['CH'], sh→['SH'], th→['TH','DH'], ng→['NG']	
		9	ai→['EY'], ee→['IY'], igh→['AY'], oa→['OW'], oo→['UH']	
		10	ar→['AA','R'], or→['AO','R'], ur→['ER'], ow→['AW'], oi→['OY']	
		11	ear→['IH','R','IY','R'], air→['EH','R'], ure→['Y','UH','R'], er→['ER']	
5-6	5	13	slz→['ZH'], wh→['W'], ph→['F']	oh, their, people, Mr, Mrs, looked, called, asked
		14	ay→['EY'], a.e→['EY'], eigh→['EY'], ey→['EY'], ei→['EY']	
		15	ea→['IY'], e.e→['IY'], ie→['IY'], ey→['IY'], y→['IY']	
		16	ie→['AY'], i.e→['AY'], y→['AY'], i→['AY']	
		17	ow→['OW'], o.e→['OW'], o→['OW'], oe→['OW']	
		18	ew→['UW'], ue→['UW'], u.e→['UW'], u→['UH'], oul→['UH']	
		19	aw→['AO'], au→['AO'], al→['AO','L'], our→['AO','R']	
		20	ir→['ER'], er→['ER'], ear→['ER']	
		21	ou→['AW'], oy→['OY']	
		22	ere→['IH','R'], eer→['IH','R'], are→['EH','R'], ear→['EH','R']	
		23	c→['K'], k→['K'], ck→['K'], ch→['K']	
		24	ce→['S'], ci→['S'], cy→['S'], sc→['S'], stl→['S'], se→['S']	
		25	ge→['JH'], gi→['JH'], gy→['JH'], dge→['JH']	
		26	le→['L','AH','L'], mb→['M'], kn→['N'], gn→['N'], wr→['R']	
		27	tch→['CH'], chlc→['SH'], ea→['EH'], a→['AA'], o→['AH']	

Table 3: Phoneme sets and progression mapping used in this study, adapted from Pearson’s Phonics Progression Chart (Pearson Education, n.d.) Each set groups grapheme-phoneme correspondences and associated target age ranges, and is assigned a phonics phase corresponding to instructional progression. The phoneme progression consists of 26 sets comprising a total of 112 grapheme-phoneme (GP) correspondences. Phase 4 (Set 12) is omitted, as it serves as a consolidation phase focusing on blending and fluency using previously introduced correspondences, rather than introducing new GP mappings. Tricky words indicate words to be learned at that phase but may contain certain graphemes that conflict with target GP mappings.

B.4 Metric 4: Maximum Phonics Level Deviation

Measures the deviation of the highest phonics phase deviation per word, averaged across the story. Note that tricky words listed in 3 were counted towards that phonics phase. Values closer to 0 indicate better alignment.

B.5 Metric 5: Mean Age of Acquisition Deviation

Measures the deviation of the average word-level Age of Acquisition (AoA) values from the target age range. Words within the target range incur no penalty, while words outside the range contribute proportionally to their distance.

$$M_5 = \delta \left(\frac{1}{|\mathcal{A}|} \sum_{w \in \mathcal{A}} A(w) \right) \quad (5)$$

$$M_4 = \left| \frac{1}{|W|} \sum_{w \in W} \max_{\ell \in L(w)} \ell - PP_t \right| \quad (4)$$

where

$$\delta(a) = \begin{cases} a_{\min} - a, & a < a_{\min} \\ 0, & a_{\min} \leq a \leq a_{\max} \\ a - a_{\max}, & a > a_{\max} \end{cases} \quad (6)$$

B.6 Metric 6: Maximum Age of Acquisition Deviation

Measures the deviation of the AoA of the most difficult word in the text from the target age range. This captures the worst-case lexical difficulty.

$$M_6 = \delta \left(\max_{w \in \mathcal{A}} A(w) \right) \quad (7)$$

B.7 Metrics 7-10: Readability

These metrics assess surface-level linguistic complexity based on sentence length, word difficulty, and character-level features. Lower complexity generally corresponds to better suitability for early readers, but readability thresholds are also presented in C.

Let:

- ASL = $\frac{\text{words}}{\text{sentences}}$ (average sentence length)
- ASW = $\frac{\text{syllables}}{\text{words}}$ (average syllables per word)
- PHW = $\frac{\text{hard words}}{\text{words}}$ (proportion of complex words)

Syllable count was obtained from ‘cmudict’ in ‘nltk.corpus’ but when the word was not found in the dataset, the ‘syllables’ library in Python was used to estimate it. The number of words whose syllable count were estimated was noted as ‘Est Syl’ in the dataset.

Metric 7: Flesch Reading Ease Measures how easy the text is to read. Higher values indicate simpler and more accessible text.

$$M_7 = 206.835 - 1.015 \cdot \text{ASL} - 84.6 \cdot \text{ASW} \quad (8)$$

Metric 8: Flesch-Kincaid Grade Level Estimates the U.S. grade level required to understand the text. Lower values indicate simpler text.

$$M_8 = 0.39 \cdot \text{ASL} + 11.8 \cdot \text{ASW} - 15.59 \quad (9)$$

Metric 9: Gunning Fog Index Measures readability based on sentence length and proportion of complex words. Lower values indicate easier readability.

$$M_9 = 0.4 \cdot (\text{ASL} + 100 \cdot \text{PHW}) \quad (10)$$

Metric 10: Automated Readability Index

Estimates readability using character-level and sentence-level features. Lower values indicate easier text.

$$M_{10} = 4.71 \cdot \frac{\text{characters}}{\text{words}} + 0.5 \cdot \frac{\text{words}}{\text{sentences}} - 21.43 \quad (11)$$

B.8 Metrics 11-12: Narrative Quality

Narrative quality is evaluated using an LLM-as-a-judge framework based on predefined rubrics (A1-4, B1-4) as stated in the prompt in Figure 11. Each criterion is scored on a 3-point scale: 0 = Not present; 1 = Partially present; 2 = Clearly present. The scores for A1-4 and B1-4 were also noted in the dataset.

Metric 11: Coherence Measures meaning-level organization, including clarity of setup, logical progression, causal relationships, and resolution.

$$M_{11} = \frac{1}{4} \sum_{k=1}^4 \frac{A_k}{2} \quad (12)$$

Metric 12: Cohesion Measures surface-level connectedness across sentences, including referential clarity and linguistic consistency.

$$M_{12} = \frac{1}{4} \sum_{k=1}^4 \frac{B_k}{2} \quad (13)$$

C Readability Thresholds

Metric	Ideal	Acceptable	Too Hard
FRE	80-100	70-80	< 60
FKGL	0-2	3-4	≥ 5
Fog	≤ 6	7-8	≥ 9
ARI	0-2	3-4	≥ 5

Table 4: Readability thresholds for early readers.

D Prompt-Level Figures

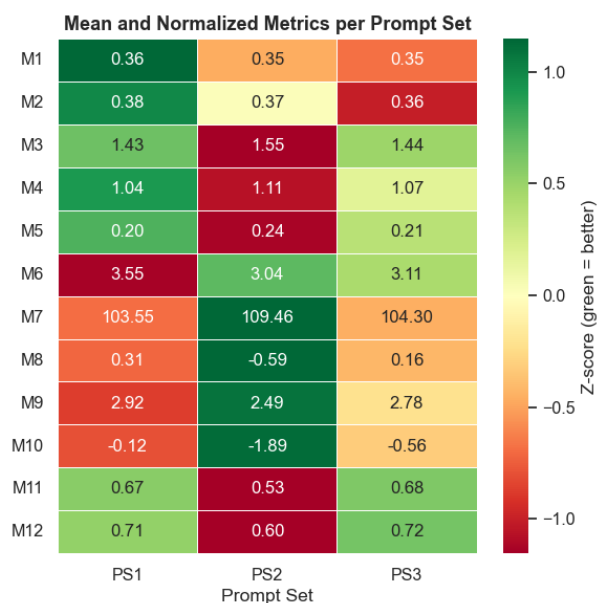


Figure 5: Performances of prompt sets across all metrics, averaged (annotation=mean) and normalized (color=Z-score) over all prompt configurations.

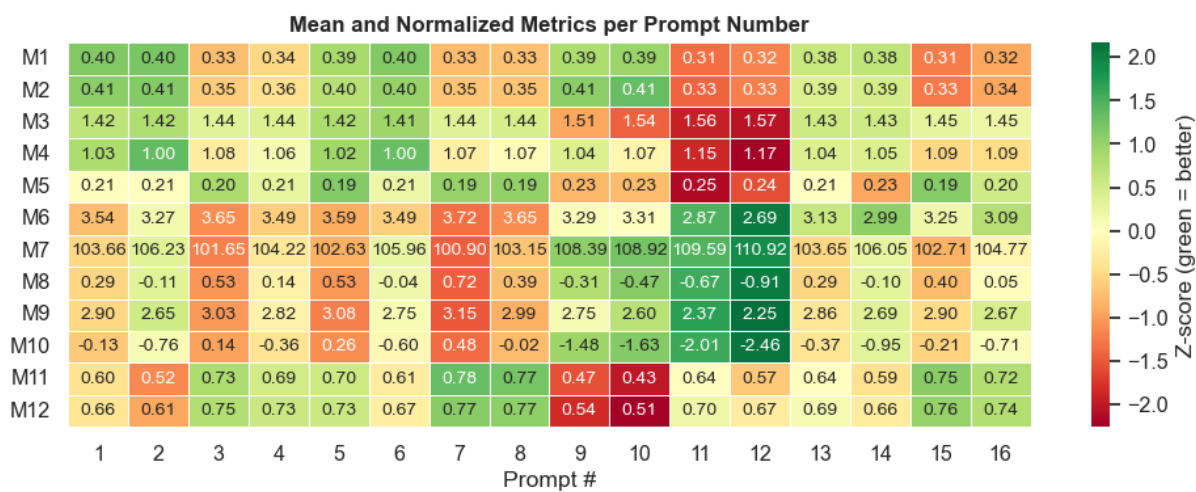


Figure 6: Performances of each specific prompt across all metrics, averaged (annotation=mean) and normalized (color=Z-score) over all prompt configurations.

E Model-Level Figures

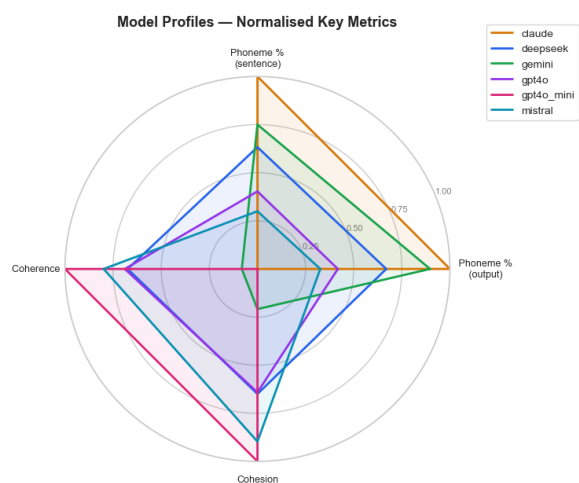


Figure 7: Model profiles for phoneme density and narrative quality show trade-offs between model performance.

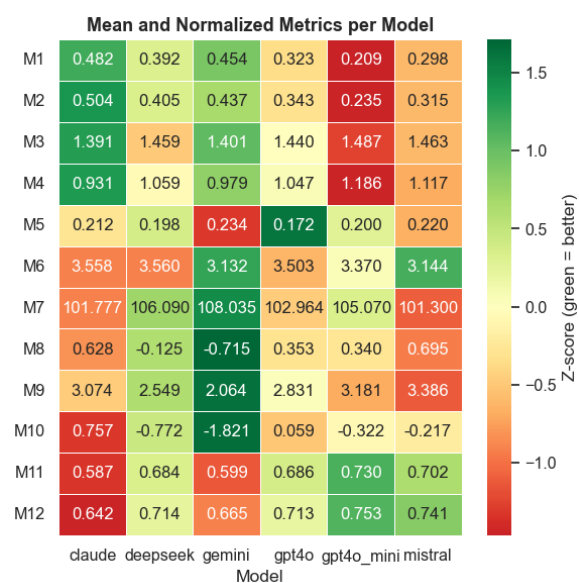


Figure 8: Performances of models across all metrics, averaged (annotation=mean) and normalized (color=Z-score) over all prompt configurations.

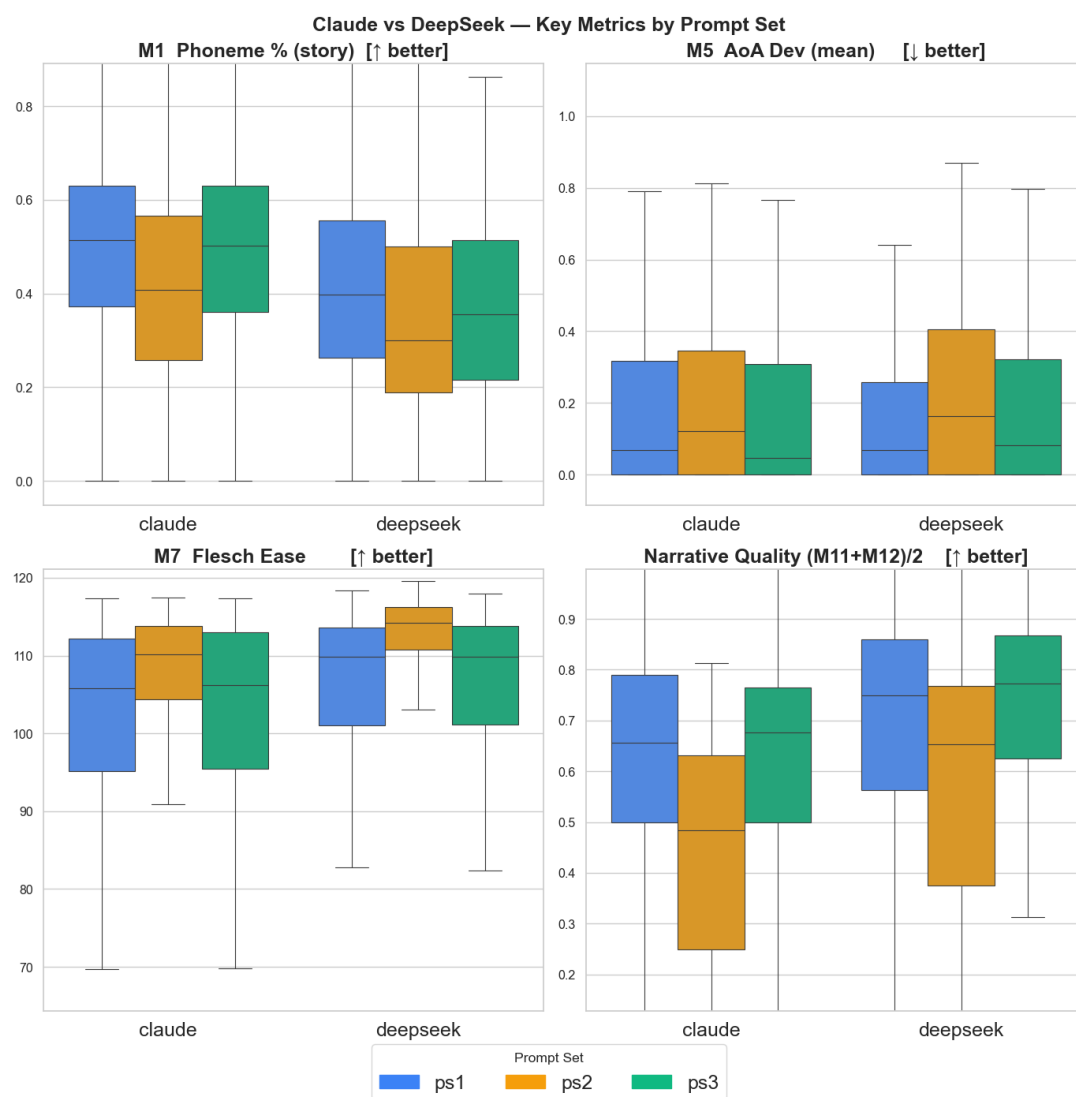


Figure 9: Performance of claude vs deepseek over metric categories using each prompt set.

F Statistical Analysis

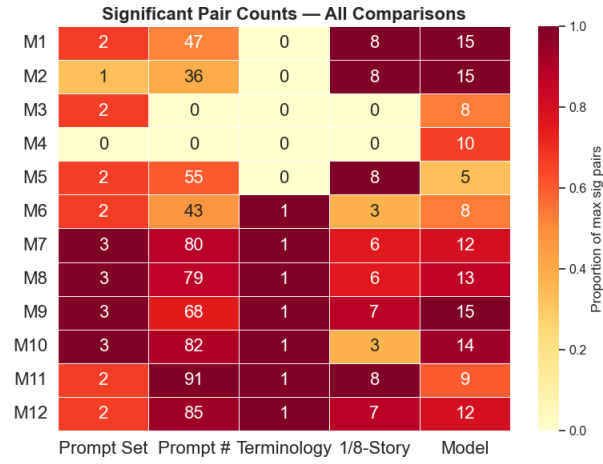


Figure 10: Count of pairs (by prompt set, prompt number, terminology, story count, and model) with significant difference ($p \leq 0.05$) across all metrics

Metric	Count	Mean	Std	Min	25%	50%	75%	Max
1. Story-Level Phoneme Density (↑)	8688	0.360	0.222	0.000	0.184	0.346	0.522	1.000
2. Sentence-Level Phoneme Density (↑)	8688	0.373	0.219	0.000	0.202	0.367	0.538	1.000
3. Mean Phonics Level Deviation (↓)	8688	1.440	0.997	0.000	0.433	2.049	2.372	3.000
4. Max Phonics Level Deviation (↓)	8688	1.053	0.728	0.000	0.310	1.094	1.666	3.000
5. Mean AoA Deviation (↓)	8688	0.206	0.275	0.000	0.000	0.085	0.328	2.394
6. Max AoA Deviation (↓)	8688	3.378	1.637	0.000	2.254	3.245	4.380	10.920
7. Flesch Reading Ease (↑)	8688	104.206	11.721	18.377	98.841	107.487	112.760	119.867
8. Flesch-Kincaid Grade Level (↓)	8688	0.196	1.681	-2.880	-0.954	-0.143	0.985	12.129
9. Gunning Fog Index (↓)	8688	2.848	1.217	0.933	2.160	2.640	3.183	21.010
10. Automated Readability Index (↓)	8688	-0.386	2.320	-8.466	-1.855	-0.448	0.966	10.723
11. Coherence (↑)	8688	0.665	0.231	0.000	0.500	0.719	0.844	1.000
12. Cohesion (↑)	8688	0.704	0.167	0.000	0.625	0.750	0.828	1.000

Table 5: Full descriptive statistics across all generated stories. Arrows indicate whether higher (↑) or lower (↓) values are generally preferable.

G Narrative Evaluation Framework

LLM Evaluation Prompt for Narrative Quality

You are an expert in children's phonics stories for speech development in early childhood education (ages 4-6). You understand that these stories are short (exactly 4 sentences), simple, and designed for early readers. You must evaluate the narrative quality of each story based ONLY on structural coherence and cohesion. Do NOT evaluate creativity, vocabulary richness, or literary sophistication. Do NOT penalize simplicity. Focus only on whether each story is logically clear and well-connected.

INSTRUCTIONS:

1. Input Requirements: - The user will provide multiple numbered phonics stories, each with a title and 4 sentences.

2. Evaluation Criteria

Evaluate each story using the following criteria.

A. COHERENCE (Meaning-Level Organization) Score each criterion: 0 = Not present 1 = Partially present 2 = Clearly present

A1. Situation Setup Clarity - Does the first sentence clearly introduce who or what the story is about?

A2. Logical Event Progression - Do the sentences follow a logical order? - Does each sentence build naturally from the previous one?

A3. Causal or Motivational Connection - Are actions connected by cause, goal, or intention? - Is it clear why events happen?

A4. Resolution or Completion - Does the final sentence complete or resolve the situation?

B. COHESION (Surface-Level Connections) Score each criterion: 0 = Not present 1 = Partially present 2 = Clearly present

B1. Referential Clarity - Are pronouns clearly linked to nouns? - Is there ambiguity in who is performing actions?

B2. Lexical Consistency - Are key words repeated or semantically related? - Is the topic maintained across all sentences?

B3. Use of Connective Devices - Are simple connectors used appropriately? (e.g., and, then, so, but)

B4. Absence of Contradiction - Are there inconsistencies or sudden unexplained changes?

3. Output Format

Return ONLY a JSON array with one object per story, in the same order as the input:

```
{
  "story": <story number>,
  "title": "<story title>",
  "coherence": {
    "A1_situation_setup": <0-2>,
    "A2_event_progression": <0-2>,
    "A3_causal_connection": <0-2>,
    "A4_resolution": <0-2>
  },
  "cohesion": {
    "B1_referential_clarity": <0-2>,
    "B2_lexical_consistency": <0-2>,
    "B3_connectives": <0-2>,
    "B4_no_contradiction": <0-2>
  }
}
```

Figure 11: Prompt used for narrative quality scoring.

H Sample of Generated Stories

#	Target GP Mapping	Prompt-Model	Story	$M_1(\uparrow)$	$M_2(\uparrow)$	$M_3(\downarrow)$	$M_4(\downarrow)$	$M_{11}(\uparrow)$	$M_{12}(\uparrow)$
1	le→['L', 'AH L']	6-claude	**Little Turtle's Rumble** Little Turtle rumbles in a puddle. The gentle turtle nibbles on a purple apple. Noble turtle tumbles while juggling a paddle. The little turtle cuddles with his uncle.	0.677	0.715	2.142	1.000	0.375	0.625
2	le→['L', 'AH L']	6-gemini	**Kyle's Little Kettle** Kyle likes to juggle apples. Purple bubbles flew from his kettle. He tickled a beetle with his little toe. Kyle giggled, "Simple, the kettle is silly!"	0.586	0.536	2.262	1.000	0.375	0.250
3	le→['L', 'AH L']	6-deepseek	**The Little Turtle's Puzzle** The little turtle waddles. He finds a purple pebble. It sparkles like a bubble. He giggles with a chuckle.	0.522	0.540	2.264	1.304	0.375	0.625
4	le→['L', 'AH L']	6-gpt4o	**Little Mice Fable** A gentle turtle found an apple. He saw little mice by the table. They played and giggled in a circle. "Share the apple," said wise turtle.	0.379	0.414	2.069	1.034	0.750	0.875
5	le→['L', 'AH L']	6-mistral	**The Little Puddle** A little puddle lay near the apple tree. The beetle and the ant took a stroll by the little puddle. The beetle said, "Let's settle near the little puddle and have a picnic with our crumble." They had a wonderful time near the little puddle, enjoying their picnic under the apple tree.	0.309	0.371	2.315	1.618	1.000	1.000
6	le→['L', 'AH L']	6-gpt4o_mini	**The Little Blue Turtle** The little blue turtle took a stroll. She saw a lovely bubble roll and roll. "Look at that bubble!" she said with glee. Then she joined it in the deep blue sea.	0.194	0.232	2.321	1.694	1.000	1.000
7	a→['AA', 'AE', 'AH']	1-claude	**Sam and Dan's Grand Plan** Sam and Dan had a plan. A cat ran past their fast car. Dan had a snack at the stand. Pat and Matt clapped at their last act.	0.909	0.918	0.197	0.212	0.250	0.250
8	a→['AA', 'AE', 'AH']	1-deepseek	**Sam and the Cat** Sam has a black cat. The cat sat on a mat. Sam pats the fat cat. The cat naps in his lap.	0.731	0.743	0.000	0.000	0.875	0.875
9	o→['AA']	14-claude	**On the Spot** Tom got a hot dog from Mom. Ron and I stop at the pond. Rob drops the mop on the cot. The frog hops on top of a rock.	0.594	0.601	0.000	0.000	0.125	0.125
10	o→['AA']	13-deepseek	**The Frog on the Log** The frog hops on the log. The dog spots the frog. The frog jumps off the log. The dog stops and sits.	0.333	0.333	0.007	0.037	1.000	0.875

Table 6: Sample stories from some models showcase tradeoffs between phonological alignment (metrics 1-4) and narrative quality (metrics 11-12). Arrows indicate whether higher ($\uparrow$) or lower ($\downarrow$) values are preferable. The full dataset of stories and scores is available at <https://huggingface.co/datasets/monicamanlises/zero-shot-phonics>

From Dialogue to Learner Modeling: Identifying Candidate Signals of Productive Use in LLM-Based Grammar Practice

Luisa Ribeiro-Flucht¹, Lanhua Huang², Xiaobin Chen¹,

¹University of Tübingen Germany, ²Guangzhou City University China,

luisa.ribeiro-flucht@uni-tuebingen.de

huanglh@gcu.edu.cn

xiaobin.chen@uni-tuebingen.de

Abstract

Adaptive language-learning systems often model progress through correctness in constrained exercises, where the target response is predefined. In dialogue-based tutors, by contrast, learners can respond appropriately in many ways, making evidence of progress harder to interpret. This raises a learner-modeling problem: determining whether learner production provides useful evidence of progress and how they might support adaptation. We address this problem using pilot data from an LLM-based dialogue system for English grammar practice, comprising 40 pre- and post-test tasks, treatment interactions, and 2,406 learner messages. We propose a coding scheme for learner production in dialogue and explore how the resulting evidence types can support adaptive decisions. Findings show that simple use traces of the target constructions were informative for predicting independent productive use in the next task, while finer-grained evidence helped interpret the robustness of learners' developing productive control. We discuss implications for adaptive grammar-based dialogue systems that use learner production to support communicative practice.

1 Introduction

With the rise of large language models (LLMs), dialogue-based computer-assisted language learning (DB-CALL; [Hou and Min, 2026](#)) has become an increasingly active area of research ([Du and Reynolds, 2025](#); [Panagiotidis, 2024](#)). For grammar practice, these systems are especially promising, as they can support interactive exchanges in which learners can encounter target constructions in context, be encouraged to use them in their own responses, and receive immediate feedback.

This makes LLM-based dialogue systems well suited to support productive grammar use, understood here as a learner's ability to use the grammatical structures of a second language to express their

own meanings ([Larsen-Freeman, 2003](#)). However, making such practice adaptive remains difficult: learners differ in what they know, how they use language, and how much practice they need ([Malovrh and Lee, 2023](#); [Ruiz et al., 2023](#); [Skehan, 1991](#)).

Adaptive dialogue-based tutors, therefore, need systematic ways to infer a learner's state from complex data and use this evidence to decide what kind of support or practice should come next ([Katz et al., 2021](#)). In structured practice environments, such as exercise-based tutoring systems, a specific response is usually expected, and performance can often be evaluated through pattern matching. In dialogue, however, a learner may either respond appropriately without using the expected grammatical pattern, use it only after support, reuse it only in a familiar context, or, ideally, produce it independently across contexts. The challenge is thus not just to detect whether the expected pattern is present or not, but to interpret what it reveals about the learner's developing grammatical control.

In this paper, we investigate this issue in the context of Aisla ([Chen et al., 2022](#)), a task-based dialogue system for targeted grammar practice. The current version of the system has been designed to model and elicit a given target construction in every turn. As a next step, we aim to support adaptive practice, which requires a better understanding of how learner progress can be modeled from interaction data. We therefore ask: what aspects of learner production in dialogue provide useful evidence of productive grammar use, and do different evidence types support different modeling goals?

Drawing on pilot data comprising 2,409 learner messages, we propose and explore a coding scheme for learner production in dialogue, examining how its categories function as candidate signals of independent productive use, partial evidence of control, and continued need for practice or support.

2 Background

Usage-based approaches to second language acquisition (SLA) view language knowledge as organized around constructions: learned form-function mappings ranging from morphemes and specific expressions to abstract grammatical patterns (Behrens, 2009; Bybee, 2006; Schmid, 2015). From this perspective, learning involves strengthening associations between forms and functions through repeated exposure and use, making constructions more accessible and stable over time (Suzuki, 2024). With growing experience, learners may also generalize beyond specific exemplars, moving from expressions such as “I walked” toward more abstract patterns such as “I [VERB]-ed” (Goldberg, 2009; Schmid, 2018).

In addition to repetition, meaningfulness has been argued to play an important role in the development of productive grammar (Lightbown, 2007; Larsen-Freeman, 2013; DeKeyser and DeKeyser, 2007). When learners use language to express meaning across situations, they are pushed beyond repeating memorized rules, toward adapting forms to communicative needs. This makes dialogue a promising setting for grammar practice as it creates opportunities to use target constructions in context. For DB-CALL systems, however, this also creates a practical challenge: If repeated, contextualized use helps build productive control, adaptive systems need principled ways to interpret learner production as evidence for pedagogical decisions.

2.1 Prior Work

Existing adaptive tutoring systems often model learners from structured evidence, such as response correctness or estimated mastery (Amaral and Meurers, 2007; Portnoff et al., 2021). Recent work has begun to incorporate dialogue interaction into learner modeling, mostly outside language learning. For example, Katz et al. (2021) describe a physics tutoring system that builds a learner model during dialogue and uses it to guide adaptive support. Borchers et al. (2024) combine dialogue-act classification with skill modeling in mathematics to examine which interaction types are associated with learning rates. In language learning, Cornillie et al. (2025) predict task completion in adaptive spoken dialogue systems from process data in functional L2 speaking tasks.

Together, these studies show that dialogue interaction can inform learner modeling and adaptation.

However, they do not directly address how learner production in grammar-focused dialogue should be interpreted. In particular, prior work offers limited guidance on how to interpret different patterns of learner production in grammar-focused dialogue, such as whether learners use the practiced grammatical pattern independently or only after support, and whether non-use still reflects meaningful participation or not.

3 System Context

The Aisla system uses LLMs to generate targeted communicative tasks in which learners are encouraged to use target constructions while engaging in meaning-focused exchange to achieve a non-linguistic goal. During these tasks, the system provides feedback with increasing explicitness, moving from recasts to explicit grammar tips, upon persistent failure to accurately use the target form given a licensed opportunity.

The system is structured over a graph-based domain model (Ribeiro-Flucht and Chen, 2026) that links forms, functions and the authentic contexts in which they can be practiced, tasks. The graph provides an explicit representation of the instructional space, allowing the system to organize practice opportunities in a way that reflects both pedagogical structure and communicative relevance.

Learner behavior is intended to update a learner-specific overlay over this graph. The purpose of that overlay is to represent the learner’s developing control over different targets and, ultimately, to support adaptive progression through the pedagogical space. Yet such updates depend on a core unresolved issue: which aspects of learner production should count as evidence.

4 Methodology

We analyze 2,409 learner messages produced by 20 adult A2-level learners of English with diverse first-language backgrounds, most commonly Chinese, Portuguese, and Spanish. Over three weeks, learners completed 20 ten-turn dialogue tasks in Aisla, targeting two English constructions: the present perfect and the present continuous. The tasks were practiced in an interleaved sequence.

The study included two task types. Treatment tasks followed a short Q&A dialogue format in which the system modeled and elicited the target construction on each turn, with incremental feedback when learners did not produce the target form.

The first two and last two tasks served as pre- and post-tests. These tasks used an information-gap format: they strongly licensed the target construction through context setting and resources, but did not model it or provide turn-level feedback. This allowed us to examine whether learners produced the target construction independently in a communicative context where the form was useful but not supplied by the system. More details on the task types and task-generation procedure are reported by Ribeiro-Flucht et al. (2025).

Learner messages were first annotated for accurate inclusion of the target construction using POLKE (Sagirov and Chen, 2025), an English Grammar Profile annotator. We then derived additional rule-based features, including recurrent target inclusion, lexical diversity, response length, and feedback uptake, defined as target inclusion immediately after feedback. An excerpt of the annotated data can be found in Appendix A. All analyses were conducted in Python 3.10. The annotated data and analysis scripts can be found in the study’s Github repository ¹.

4.1 Evidence Classification

To examine how learner production may inform future adaptive decisions, each learner message was assigned to one of five mutually exclusive evidence categories. The categories were designed as a theoretically motivated operationalization of learner evidence in dialogue. They draw on three distinctions that are central to learner modeling for productive use: whether the target construction is produced at all, whether production occurs independently or with immediate support, and whether responses without the target construction nevertheless provide evidence of meaningful task engagement.

The distinction between independent and scaffolded production is motivated by research on corrective feedback, where learner responses following feedback are treated as informative but not equivalent to unassisted production (Lyster and Ranta, 1997). The distinction between initiated and sustained independent use further separates first retrieval of a target construction from continued use once the construction has become active in the dialogue (Ellis et al., 2015; Pannell et al., 2017). Finally, the distinction between engaged responses not containing the target construction and underuse reflects the ambiguity of non-use in

open-ended dialogue: a learner may respond meaningfully without using the target, or they may provide a minimal response that offers little evidence for learner modeling. We therefore propose the following categories:

- **Independently initiated:** the learner produced the target construction without scaffolding after a previous learner message that did not contain the target. This was treated as evidence that the learner could initiate use of the construction without immediate support.
- **Independently sustained:** the learner produced the target construction without scaffolding after having previously used it. This was treated as evidence that the learner could continue using the construction once it had become active in the dialogue.
- **Scaffolded:** the learner produced the target construction after receiving support from the system, such as a recast or a grammar tip. This was treated as partial evidence of productive control, since the learner used the form but did so with immediate support.
- **Engaged non-target:** the learner did not produce the target construction, but gave a response with more than two words. This was treated as evidence of task engagement without target-form use.
- **Underuse:** the learner did not produce the target construction and responded with two words or fewer. This was treated as the weakest evidence for productive control, since the response provided little opportunity to observe target-form use.

In addition, we tracked lexical diversity as the cumulative number of different lemmas used within the target construction. As discussed in Section 2, from a usage-based perspective, varied lexical content in a construction may indicate more robust productive control than repeated use of a single exemplar, making lexical diversity a complementary signal of constructional development.

4.2 Task-Level Prediction

Our first analysis examined whether prior learner evidence could predict independent target use in the next task. This analysis was motivated by the adaptive use case of the learner model: at each task

¹https://github.com/luisards/productive_grammar_signals

boundary, the system needs evidence for deciding whether a learner is likely to use the target construction independently, or whether further practice and support may be needed. We therefore treated next-task independent use as a short-term prediction problem.

The unit of analysis was the learner-task pair. For each of the 20 learners, performance on tasks 2-20 was predicted from all preceding interaction data, yielding 380 observations. Because this was a small pilot dataset, the two target constructions were pooled for this analysis rather than modeled separately. This allowed us to estimate whether the proposed evidence encodings carried predictive signal across target constructions, while leaving construction-specific analyses to future work with larger samples.

The outcome was binary: whether the learner produced the target independently at least once during the task, defined as any turn classified as *Independently initiated* or *Independently sustained*. At each task boundary, features were computed from all prior turns as either cumulative or recency-weighted proportions, with exponential decay ($\lambda = 0.85$). *Underuse* served as the reference category.

We compared seven nested logistic regression models: an intercept-only baseline, binary target-use traces, typed evidence profiles, and typed profiles with lexical diversity, in cumulative or recency-weighted form where applicable. This allowed us to test whether richer evidence representations improved on a simple target-use trace and whether lexical diversity added predictive value. Models were evaluated with leave-one-user-out cross-validation using AUC-ROC, accuracy, balanced accuracy, and log-loss.

All models used L2-regularized logistic regression ($C = 1.0$). For recency-weighted models, decay sensitivity was also tested across $\lambda \in \{0.75, 0.80, 0.85, 0.90, 0.95\}$. Coefficients from typed models fit on the full sample were used to interpret the relative contribution of each evidence type, including scaffolded use, engaged non-target responses, and lexical diversity.

4.3 Post-test Prediction

A second analysis examined which encoding best predicted independent target use on the post-treatment probe tasks, our closest measure of less scaffold-dependent productive ability. These probes were information-gap communicative tasks with no modeling of the target form and no feed-

back.

The unit of analysis was the learner-construction pair, yielding 40 post-tests (20 learners $\times$ 2 constructions). The outcome was binary: whether the learner produced the target independently on the post-test probe task for that construction (task 19 for present perfect, task 20 for present continuous).

Predictors were derived from the treatment phase only (tasks 3-18) for the corresponding construction. These included the binary target-use rate, the rates of each evidence type, and lexical diversity. Because the sample was too small for reliable multivariate logistic regression, each feature was evaluated separately using AUC-ROC. This allowed a direct comparison of simple binary traces, typed evidence measures, and lexical diversity as candidate signals of post-treatment productive use.

4.4 Results

Across all learner messages, non-target responses were most common. *Underuse* accounted for 33.2% of messages and *engaged non-target* responses for 32.0%. Among target-containing messages, *independently sustained* use was most frequent (14.2%), followed by *independently initiated* use (11.5%) and *scaffolded* use (9.0%). At the task level, learners produced the target independently in 68.2% of tasks overall, although this varied widely across tasks (30%-95%) and across learners.

4.5 Task-Level Model Comparison

Table 1 reports leave-one-user-out cross-validation results. All evidence-based models outperformed the intercept-only baseline ($\Delta\text{AUC} > .23$), but differences among the substantive models were modest. The best performance was obtained by the recency-weighted typed model with lexical diversity (M6, AUC = .751), followed closely by the cumulative and recency-weighted binary models (M1: .746; M2: .741). The main finding is that a simple binary trace of prior target use captured most of the predictive signal. Typed evidence profiles did not improve performance on their own, slightly underperforming the binary models in both cumulative and recency-weighted form. Their main contribution was therefore interpretive rather than predictive.

At the same time, coefficients from the typed models revealed a clear evidence hierarchy (Table 2): scaffolded production was the strongest positive predictor of later independent use, followed by *Independently sustained* and *Independently ini-*

Model	k	AUC	Acc.	BAcc.	Log-loss
M0: Intercept only	0	.500	.682	.500	.626
M1: Binary cumulative	1	.746	.671	.523	.548
M2: Binary recency	1	.741	.703	.608	.541
M3: Typed cumulative	4	.736	.682	.500	.569
M4: Typed recency	4	.732	.682	.500	.555
M5: Typed cumulative + diversity	5	.749	.689	.576	.547
M6: Typed rec. + diversity	5	.751	.705	.608	.538

Table 1: LOUO-CV results for all seven models.

tiated production, while engaged non-target responses were close to neutral relative to underuse. Adding lexical diversity produced small but consistent gains, and cumulative and recency-weighted encodings performed similarly overall.

Rank	Evidence type	β	OR
1	Scaffolded	+2.978	19.65
2	Independently sustained	+2.739	15.47
3	Independently initiated	+1.166	3.21
4	Engaged non-target	+0.157	1.17

Table 2: Logistic regression coefficients (β) and odds ratios (OR) for typed evidence categories. Coefficients are interpreted relative to underuse, which served as the reference category.

4.6 Post-test Validation

Across all probe task observations, independent target production increased from 62.5% on the pre-probe to 80.0% on the post-probe. Table 3 reports the predictive value of each treatment-phase feature for post-probe independent production in terms of area under the curve (AUC) and p -value.

Feature	AUC	p
Lexical diversity	.979	< .001
Binary target rate	.959	< .001
Independently initiated	.957	< .001
Scaffolded rate	.893	< .001
Independently sustained	.877	.001
Engaged non-target	.510	.934
Underuse	.049	< .001

Table 3: Predictive value of features for post-test independent target production.

Lexical diversity was the strongest predictor (AUC = .979, $\rho = +.665$, $p < .001$), outperforming the binary target rate (AUC = .959, $\rho = +.637$, $p < .001$). Unlike in the task-level analysis, where it added only modest value, it was highly informative here, suggesting that lexical variation is more closely related to broader productive control than to next-task performance.

The remaining binary and typed measures were similar in predictive power, again indicating that a simple target-use trace captures most of the useful signal. Among non-target measures, underuse was a strong negative predictor ($p < .001$), whereas engaged non-target was uninformative ($p = .934$). The two constructions also differed: present perfect improved from 30% to 70%, while present continuous remained near ceiling (95% to 90%). Treatment-phase evidence was therefore more informative for present perfect, where there was greater room for measurable change.

4.7 Conclusion

This study examined how learner behavior in LLM-based L2 grammar practice can inform learner modeling. The findings suggest that different evidence types serve different purposes: prior target use was most useful for predicting immediate next-task performance, while finer-grained evidence helped characterize the robustness of developing productive control. Scaffolded production emerged as a strong positive signal, suggesting that system modeling and timely clarification may help make the practiced pattern available for use. Lexical diversity appeared to capture broader control over that pattern, pointing to the value of grammar practice with varied lexical fillers.

These findings suggest that less constrained communicative grammar practice may be made systematically tractable for adaptive decisions. Rather than treating learner production as too noisy to support principled adaptation, the results indicate that it can be encoded in ways that are both practically usable and theoretically meaningful. These findings are especially important for language learning, where development is gradual and linguistic knowledge is strengthened and extended through use.

Limitations

This study should be interpreted in light of several limitations. First, the sample was small, which limits the strength and generalizability of the conclusions. Second, the analysis could not fully address cross-context tracking or cleanly separate item-specific reuse from more general productive control, beyond the lexical diversity measure. Third, the logistic regression treated observations as independent rather than modeling tasks as nested within learners, which would be preferable in a larger dataset. Finally, three learners showed ceiling-level target use across tasks, leaving little variation to explain at the individual level.

Future work should test these patterns in larger samples and with more lexically open tasks, especially to clarify whether scaffolded success reflects emerging generalization or continued dependence on practiced forms.

References

- Luiz Amaral and Detmar Meurers. 2007. [Conceptualizing student models for ICALL](#). In *User Modeling 2007: Proceedings of the Eleventh International Conference*, Lecture Notes in Computer Science, Wien, New York, Berlin. Springer.
- Heike Behrens. 2009. [Usage-based and emergentist approaches to language acquisition](#). *Linguistics*, 47(2):383–411.
- Conrad Borchers, Kexin Yang, Jionghao Lin, Nikol Rummel, Kenneth R. Koedinger, and Vincent Aleven. 2024. [Combining dialog acts and skill modeling: What chat interactions enhance learning rates during ai-supported peer tutoring?](#) In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 117–130, Atlanta, Georgia, USA. International Educational Data Mining Society.
- Joan Bybee. 2006. [From usage to grammar: The mind's response to repetition](#). *Language*, 82(4):711–733.
- Xiaobin Chen, Elizabeth Bear, Bronson Hui, Haemant Santhi-Ponnusamy, and Detmar Meurers. 2022. Education theories and ai affordances: Design and implementation of an intelligent computer assisted language learning system. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium*, pages 582–585, Cham. Springer International Publishing.
- Frederik Cornillie, Julie Gijpen, Sameh Said-Metwaly, Steffen Luybaert, and Wim Van Den Noortgate. 2025. [Toward adaptive spoken dialogue systems for language learning: Predicting task completion from learning process data](#). *CALICO Journal*, 42(3):413–436.
- Robert M. DeKeyser and Robert DeKeyser. 2007. [Introduction: Situating the concept of practice](#), page 1–18. Cambridge Applied Linguistics. Cambridge University Press.
- Yunfei Du and Barry Lee Reynolds. 2025. [Rooted in and beyond interaction: A systematic review of interactive affordances of chatbots for language learning amidst the rise of large language models](#). *Acta Psychologica*, 259:105307.
- Nick C. Ellis, Matthew Brook O'Donnell, and Ute Römer. 2015. [Usage-Based Language Learning](#), chapter 7. John Wiley & Sons, Ltd.
- Adele E. Goldberg. 2009. [The nature of generalization in language](#). *Cognitive Linguistics*, 20(1):93–127.
- Zhuohan Hou and Shangchao Min. 2026. [Dialogue-based computer-assisted language learning systems for second language speaking development: A three-level meta-analysis](#). *ReCALL*, 38(1):40–56.
- Sandra Katz, Patricia Albacete, Irene-Angelica Chounta, and 1 others. 2021. [Linking dialogue with student modelling to create an adaptive tutoring system for conceptual physics](#). *International Journal of Artificial Intelligence in Education*, 31:397–445.
- Diane Larsen-Freeman. 2003. [Teaching Language: From Grammar to Gramaring](#). Thomson/Heinle, Boston.
- Diane Larsen-Freeman. 2013. [Transfer of learning transformed](#). *Language Learning*, 63(s1):107–129.
- Patsy Martin Lightbown. 2007. [Transfer appropriate processing as a model for classroom second language acquisition](#). In ZhaoHong Han, editor, *Understanding Second Language Process*, pages 27–44. Multilingual Matters, Bristol and Blue Ridge Summit.
- Roy Lyster and Leila Ranta. 1997. [Corrective feedback and learner uptake: Negotiation of form in communicative classrooms](#). *Studies in Second Language Acquisition*, 19(1):37–66.
- Paul A. Malovrh and James F. Lee. 2023. [The effects of prior knowledge on explicit rule formation among 12 spanish learners receiving structured input through guided-inductive instruction](#). *Ampersand*, 10:100116.
- Panagiotis Panagiotidis. 2024. [LLM-based chatbots in language learning](#). *European Journal of Education*, 7(1):102–123.
- Justin Pannell, Friederike Partsch, and Nicholas Fuller. 2017. [The output hypothesis: From theory to practice](#). *TESOL Working Paper Series*, 15:126–159.
- Lucy Portnoff, Erin Gustafson, Joseph Rollinson, and Klinton Bicknell. 2021. [Methods for language learning assessment at scale: Duolingo case study](#). *International Educational Data Mining Society*.

- Luisa Ribeiro-Flucht and Xiaobin Chen. 2026. [The L2 network: A CEFR-aligned knowledge graph for grammar domain modeling](#). In *Proceedings of the Workshop on Structured Linguistic Data and Evaluation*, pages 148–159, Palma de Mallorca, Spain. ELRA Language Resources Association.
- Luisa Ribeiro-Flucht, Xiaobin Chen, and Detmar Meurers. 2025. [A framework for proficiency-aligned grammar practice in LLM-based dialogue systems](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 978–987, Vienna, Austria. Association for Computational Linguistics.
- Simón Ruiz, Patrick Rebuschat, and Detmar Meurers. 2023. [Supporting individualized practice through intelligent CALL](#). In Yuichi Suzuki, editor, *Practice and Automatization in Second Language Research: Perspectives from Skill Acquisition Theory and Cognitive Psychology*, pages 119–143. Routledge.
- Nelly Sagirov and Xiaobin Chen. 2025. [POLKE: A system for comprehensively annotating pedagogically-oriented grammatical structure use in language production](#). Preprint.
- Hans-Jörg Schmid. 2015. [A blueprint of the entrenchment-and- conventionalization model](#). *Yearbook of the German Cognitive Linguistics Association*, 3(1):3–26.
- Hans-Jörg Schmid. 2018. [Unifying entrenched tokens and schematized types as routinized commonalities of linguistic experience](#). *Yearbook of the German Cognitive Linguistics Association*, 6(1):167–182.
- Peter Skehan. 1991. [Individual differences in second language learning](#). *Studies in Second Language Acquisition*, 13(2):275–298.
- Yuichi Suzuki. 2024. [Skill acquisition theory: Learning-to-use and usage-for-learning SLA](#). In *Usage in Second Language Acquisition*, pages 147–168. Routledge.

A Appendix

(a) Present perfect – Learner 95, treatment task 9

#	Message	Evidence type
1	S: Hello! What is something you have always wanted to learn?	
2	L: to ride a bike	ENGAGED NON-TARGET
3	S: Cool! I have always wanted to ride a bike, too. Have you ever tried to learn before, or is it something you have always wished to do but never started?	
4	L: i <u>have tried</u> to learn but i failed	INDEP. INITIATED
5	S: I see. Where have you practiced riding a bike?	
6	L: i <u>haven't practiced</u> since 2015	INDEP. SUSTAINED
7	S: Oh, ok. What is something else you have always wanted to try?	
8	L: i <u>have always wanted</u> to travel abroad	INDEP. SUSTAINED
9	S: That is a wonderful wish. Which country have you always dreamed of visiting the most?	
10	L: probably canada	UNDERUSE
11	S: I have visited Canada before. I hope you get to visit there one day, too. Talk to you later!	

(b) Present continuous – Learner 22, treatment task 12

#	Message	Evidence type
1	S: Hello. I am always watching movies in my free time. What is something that you are always doing?	
2	L: I'm <u>always watching</u> short videos.	INDEP. INITIATED
3	S: I am usually working on my computer in the evenings. What are you usually doing at that time?	
4	L: I'm <u>usually doing</u> my homework and reading books in the evenings	INDEP. SUSTAINED
5	S: That sounds productive. I am often reading in the evenings too. What kind of books are you reading these days? ^r	
6	L: I'm <u>reading</u> a book by a Chinese female writer. The words of the book are very delicate.	SCAFFOLDED
7	S: That sounds interesting. I am currently reading a novel too. What is the writer's name? ^r	
8	L: Soy tea.	UNDERUSE
9	S: Oh, you mean Su Tong? I am enjoying his writing style as well. [...] ^r	
10	L: I think so. See you later!	ENGAGED NON-TARGET
11	S: Yes, I am thinking his stories are very powerful. Talk to you later!	

Figure 1: Annotated dialogue excerpts illustrating the five evidence types. (a) The “Discussing long-standing wishes” task, targeting the present perfect affirmative. (b) The “Describing habits” task, targeting the present continuous affirmative form.

Evaluating Adaptive Personalization of Educational Readings with Simulated Learners

Ryan T. Woo* Anmol Rao* Aryan Keluskar Yinong Chen

School of Computing and Augmented Intelligence
Arizona State University
{rtwoo, arao75, akeluska, yinong.chen}@asu.edu

Abstract

We present a framework for evaluating adaptive personalization of educational reading materials with theory-grounded simulated learners. The system builds a learning-objective and knowledge-component ontology from Wikibooks open textbooks, curates it in a browser-based Ontology Atlas, labels textbook chunks with ontology entities, and generates aligned reading–assessment pairs. Simulated readers learn from passages through a Construction–Integration-inspired memory model with DIME-style reader factors, KREC-style misconception revision, and an open New Dale–Chall readability signal. Answers are produced by score-based option selection over the learner’s explicit memory state, while BKT drives adaptation. Under this explicit simulator parameterization and three sampled subject ontologies, adaptive reading significantly improved outcomes in computer science, yielded smaller positive but inconclusive gains in inorganic chemistry, and was neutral to slightly negative in general biology.

1 Introduction

With more students entering higher education, instructors increasingly face a familiar mismatch: students are expected to learn the same material at the same pace even though their understanding of particular concepts can vary widely. In an introductory course, one student may already understand a subset of the target concepts, while another may still hold robust misconceptions about those same concepts. When both students are given identical readings, examples, and explanations, one may be under-challenged while the other is overwhelmed. Addressing this in a real classroom typically requires instructors to rewrite materials, manually identify which concepts students struggle with, and

provide individualized support. Those interventions are difficult to scale.

Intelligent tutoring systems (ITSs) aim to address this problem by adapting instruction to the learner. Recent reviews suggest that ITSs often improve learning and performance, but that their impact depends strongly on the instructional setting, the adaptation mechanism, and the way systems are evaluated (Huang et al., 2025; L  tourneau et al., 2025). In many ITS settings, however, the core intervention is an item sequence, a hint policy, or post-hoc feedback. Our setting is different: the primary intervention is *reading material*. Questions are still important, but mainly because they reveal evidence about what the learner knows after reading and provide observations for mastery tracking.

To evaluate whether adaptive reading policies help, we need simulated learners that can (i) learn from text, (ii) differ in reading-related characteristics, (iii) preserve misconceptions rather than merely lacking knowledge, and (iv) later answer multiple-choice questions from what they remember rather than from unrestricted access to the source passage. Simulated learners have long been proposed as tools for teachers, for learners, and for instructional designers conducting formative evaluation (VanLehn et al., 1994). More recently, a systematic review across AIED-related fields found uses spanning teacher training, adaptive algorithm development, virtual collaborators, and testing of learning environments, while also concluding that realism and validity are often under-tested (K  ser and Alexandron, 2024). Off-the-shelf LLM prompting is a poor fit for that requirement because powerful LLMs often answer beyond the intended ability of a partially knowledgeable learner. Yuan et al. (2026) characterize this problem as a *competence paradox* and argue for constrained student simula-

*equal contribution.

tion under an explicit epistemic state specification.

We therefore present a theory-grounded simulated reader for adaptive reading personalization. The system builds a learning-objective (LO) and knowledge-component (KC) ontology from open textbooks, tracks KC mastery with BKT (Corbett and Anderson, 1994), generates ontology-driven reading and assessment variants, and simulates reading via a comprehension model grounded in the Construction–Integration (CI) theory of discourse comprehension (Kintsch, 1988). Learner heterogeneity is parameterized with DIME-style reader factors (Cromley and Azevedo, 2007; Cromley et al., 2010), misconceptions are revised through the Knowledge Revision Components framework (KREC) (Kendeou and O’Brien, 2014; Kendeou, 2024), and later answer selection is performed through score-based option selection over an explicit epistemic boundary.

The experiments focus on offline simulation so that the instructional policy can be studied before classroom deployment. This setup compares adaptive and non-adaptive reading policies under matched simulated cohorts while keeping the content pipeline, question format, and mastery tracker fixed.

Our main contributions are as follows:

- We formulate adaptive reading personalization as a closed-loop problem in which reading is the primary intervention and assessment is the observation channel for KC-level mastery tracking.
- We present an ontology-grounded content pipeline that links open textbook chunks, reading variants, and assessment items to shared LOs and KCs.
- We develop a simulated reader that combines CI, DIME, KREC, and an open New Dale–Chall readability signal to model what a learner comprehends, revises, retains, and later retrieves.
- We report multi-subject experimental results showing that adaptive reading yields domain-dependent gains: statistically reliable improvements in the sampled computer-science setting, positive but inconclusive gains in inorganic chemistry, and neutral to slightly negative outcomes in general biology.

2 Related Work

2.1 Adaptive tutoring and knowledge tracing

Knowledge tracing estimates latent mastery from observed responses, and BKT remains attractive because it is interpretable at the level of individual KCs (Corbett and Anderson, 1994). Although deep models can improve prediction, interpretability remains important for deployment and diagnosis (Ding and Larson, 2021). We therefore use BKT so that the state driving adaptation remains understandable.

Most ITS personalization focuses on problem selection, hints, or feedback. By contrast, we center on explanatory reading material. Because reading can change knowledge without explicit responses, we separate the learner’s hidden *true* knowledge from the tutor’s BKT-based *estimate*, and only the latter drives adaptation.

2.2 Simulated learners in educational technology

Simulated learners have long been used in educational technology. VanLehn et al. (1994) categorize their use for teachers, learners, and instructional designers; our setting aligns most closely with formative evaluation of instructional policies before deployment.

Recent work highlights broader uses, including teacher training, hypothesis generation, adaptive algorithm development, virtual collaborators, and environment testing, while also noting that many simulators remain under-validated for their intended decisions (Käser and Alexandron, 2024). This motivates our emphasis on explicit learner state, inspectable memory, and theory-grounded reading dynamics.

2.3 Reading comprehension and computational reader models

Our model is grounded in the Construction–Integration framework (Kintsch, 1988), where readers form a propositional *textbase* and a richer *situation model* integrating prior knowledge. This supports modeling partial retention, revision, and distortion of information.

We extend CI with DIME, which captures reader differences in background knowledge, vocabulary, and inference (Cromley and Azevedo, 2007; Cromley et al., 2010), and KREC, which models revision of incorrect prior knowledge (Kendeou and O’Brien, 2014; Kendeou, 2024).

This matters because many systematic wrong answers reflect stable misconceptions rather than random noise (van den Broek and Kendeou, 2008; Gierl et al., 2017).

Prior work has implemented CI-like automated readers (Lemaire et al., 2006); our contribution is to embed such a reader in a KC-level adaptive instructional loop.

2.4 Epistemic boundaries and answer modeling

Recent work argues that valid student simulation should expose an explicit epistemic state and bounded competence (Yuan et al., 2026). We adopt that principle, but do not use free-form generation for answer choice. Instead, answers are selected by scoring options from the learner’s memory state, including mastery, misconceptions, retrieved traces, and item difficulty. This design is efficient, auditable, and consistent with constrained learner behavior. It also favors inspectable state and reproducible repeated simulations over unconstrained generative answers.

2.5 Readability and reader–text matching

Reader–text fit is important but should remain simple to operationalize. We use the New Dale–Chall readability formula (Chall and Dale, 1995), computed via textstat (Bansal and Aggarwal, 2026), as a reproducible measure of text difficulty combined with learner parameters. The formula is treated only as a lightweight signal; its tradeoffs are discussed in the limitations.

3 System Overview

The overall system converts open educational content into a grounded personalization loop. The pipeline can be summarized in five stages: (1) ontology construction and authoring, (2) chunk labeling and retrieval indexing, (3) ontology-driven reading and assessment generation, (4) theory-grounded simulated reading and score-based answer selection, and (5) BKT-based adaptation. Figure 1 separates the fixed source-corpus, ontology, grounding, and generation stages from the online simulation. Figure 2 shows the simulated reading and adaptation loop as a closed cycle.

The experiments use Wikibooks source textbooks: *Foundations of Computer Science*, *General Biology*, and *Introduction to Inorganic Chemistry* (Wikibooks contributors, 2026a,b,c). These books

serve as course-specific corpora: the ontology, retrieval index, generated readings, and assessments are grounded in the same instructional source material.

3.1 Ontology Atlas and ontology construction

We first build an ontology of learning objectives and knowledge components from open textbooks. Each LO maps to one or more KCs. The ontology is created with an LLM-assisted workflow that proposes candidate LOs and KCs from source material and then consolidates them into a course-specific YAML schema. We use the ontology as a common representational layer for retrieval, generation, mastery tracking, and simulation.

The ontology construction process is semi-automated rather than fully automatic. LLM-generated LO and KC candidates are treated as proposals that can be accepted, merged, split, renamed, or deleted in Atlas before experiment artifacts are generated. Merge decisions are appropriate when two KCs represent the same assessable concept at the same grain size; split decisions are appropriate when a proposed KC combines multiple independently assessable skills or concepts. In the reported workflow, ontology files are fixed within each subject-level adaptive-control comparison so that the two policies use the same ontology.

To make this layer inspectable, we built Ontology Atlas, a browser-based ontology workspace backed directly by the YAML files used elsewhere in the system. Atlas exposes a coverage inventory showing chapter, LO, and KC counts; a searchable navigator over IDs, statements, labels, and descriptions; a D3 relationship graph for the chapter → LO → KC hierarchy; and a details panel for editing entries, importing/exporting YAML, validating fields, and saving back to disk. Atlas is therefore not just a visualization layer; it is the authoring interface for the ontology that the generation and experiment services consume.

In the current codebase, Ontology Atlas surfaces multiple full-course ontologies, including the three used in the experiments reported here: foundations of computer science (16 chapters, 53 LOs, 131 KCs), general biology (20 chapters, 60 LOs, 172 KCs), and introductory inorganic chemistry (12 chapters, 57 LOs, 177 KCs). This coverage view was useful when selecting sampled LOs for the experiments and when checking whether KC granularity was comparable across subjects.

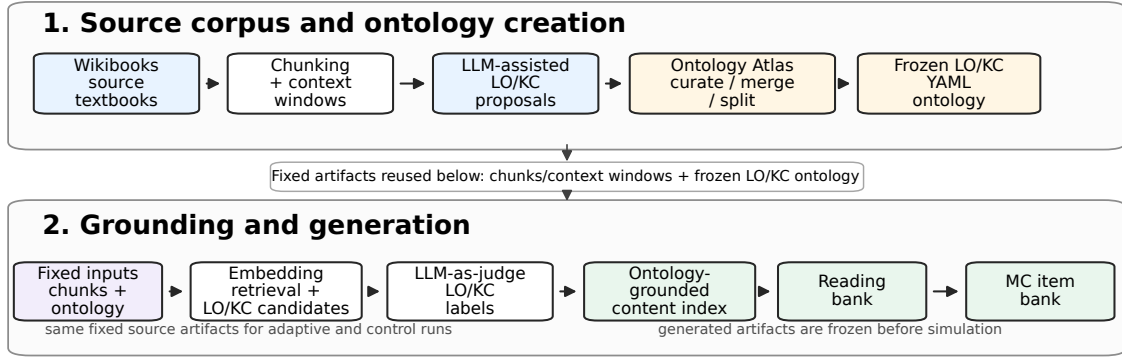


Figure 1: Fixed source-corpus, ontology, grounding, and generation pipeline used by both conditions.

3. Simulated reading and adaptation loop

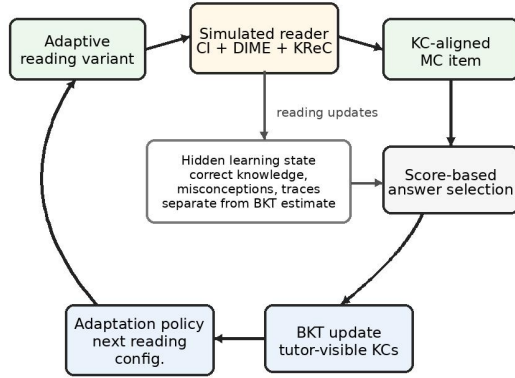


Figure 2: Closed simulated reading and adaptation loop.

3.2 Chunk labeling and retrieval indexing

The textbook corpus is segmented into fine-grained chunks and coarser context chunks. Fine-grained chunks serve as the primary evidence units for labeling, while coarser chunks provide broader context during downstream generation. Each fine-grained chunk is labeled with one or more LOs and KCs using a three-stage LLM-as-judge pipeline:

1. encode the chunk as a semantic embedding,
2. retrieve candidate LO and KC labels with nearest-neighbor search in a vector database, and
3. prompt an LLM to adjudicate among the candidates and return the final label set.

The result is a grounded index in which every retained content unit is tied to the ontology and remains linked to its surrounding context. Because Atlas edits the same YAML that the labeling and

generation services read, ontology revisions immediately propagate to retrieval and generation.

3.3 Ontology-driven reading and assessment generation

Using the labeled corpus, we generate paired readings and assessments that comprehensively cover one LO at a time. Each generated reading is grounded in retrieved fine-grained chunks and optional coarser context chunks; generation parameters control explanation depth, example density, misconception refutation, and target difficulty. Each assessment is a fresh isomorphic multiple-choice variant aligned to the same LO and KCs rather than an exact reuse of the same item surface form.

This choice is important for two reasons. First, exact question reuse would make policy evaluation more sensitive to item wording than to learner state. Second, our simulated learners do not learn from assessment interactions; only reading updates their hidden semantic state. Using fresh isomorphic variants therefore preserves comparability across iterations without allowing assessment exposure to become an unintended instructional channel.

For preprocessing, semantic embeddings used qwen3-embedding-8b; LLM-assisted ontology proposal, chunk-label adjudication, reading generation, and assessment generation used qwen3-30b-a3b-instruct-2507. Once generated, the ontology, labels, readings, and assessments are treated as fixed artifacts for the adaptive-control comparison.

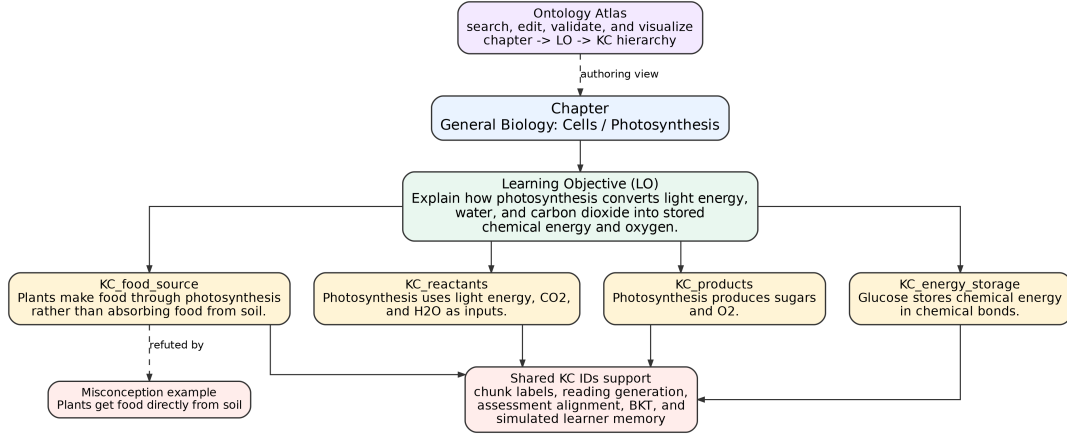


Figure 3: Example chapter → LO → KC relation visualized by Ontology Atlas for photosynthesis.

4 Methodology

4.1 Separating hidden learning from observed mastery

The simulated learner maintains two distinct states. The first is a hidden *true learning state* that governs what the learner actually knows, misunderstands, and remembers. The second is the tutor-visible BKT estimate that is updated only from observed answers. This separation prevents the simulation from becoming self-fulfilling: the personalization policy acts on what the tutor *infers*, not on hidden ground truth.

For learner i and KC k , we represent hidden state as

$$z_i = \{m_{i,k}^*, w_{i,k,1:M}, r_i, s_i, e_i\}, \quad (1)$$

where $m_{i,k}^*$ is the strength of correct knowledge for KC k , $w_{i,k,1:M}$ are strengths of KC-specific misconceptions, r_i is a reader profile, s_i is a revision-and-response profile, and e_i stores recent episodic traces from readings. The tutor only observes a separate KC mastery estimate $\hat{m}_{i,k}$ maintained by BKT.

4.2 Reader profiles and readability matching

We do not use vague “learning styles.” Instead, learner heterogeneity is parameterized through theory-grounded reader profiles. Following DIME, each learner has parameters for background knowledge, vocabulary, inferencing, and strategy use (Cromley and Azevedo, 2007; Cromley et al., 2010). Because the cohort is intended to approximate college freshmen who have completed secondary school, basic decoding is assumed to be near ceiling; the meaningful variability is in

comprehension-relevant factors and in misconception prevalence.

We combine these learner-side parameters with a reader–text fit term based on the New Dale–Chall score. The score is used as a compact, reproducible difficulty signal alongside the richer learner-side profile.

For passage j , let d_j be the readability score computed with `textstat`. Let a_i be a scalar readability-ability parameter for learner i . We define

$$\text{match}_{ij} = 1 - \frac{|a_i - d_j|}{6}, \quad (2)$$

clipped to $[-1, 1]$, so higher values indicate a better match between learner and text. This term does not replace the richer reader profile; it provides a compact difficulty signal that can be integrated into the comprehension update.

4.3 From passages to proposition-like teaching events

We model reading at the level of comprehension and memory, not time. Each generated passage is segmented into short teaching events, which serve as proposition-like units for the simulation. The current implementation uses a lean operational schema rather than full semantic parsing: each event stores its surface text, linked KC IDs, a clarity score, a refutation-strength score, and a refresh flag.

These fields have direct roles in the simulation. *KC links* determine which latent mastery variables and misconception weights the event can update. *Clarity* approximates how easy the event is to encode; in the current implementation it is estimated from event length, so shorter events are treated

State component	Meaning	Role in the simulation
$m_{i,k}^*$	Hidden strength of correct knowledge for KC k	Governs durable support for correct answers and future learning from related readings
$w_{i,k,m}$	Hidden strength of misconception m for KC k	Attracts misconception-consistent distractors and resists revision when refutation is weak
r_i	Reader profile (background knowledge, vocabulary, inferencing, strategy use, readability ability)	Determines how well teaching events are encoded from text
s_i	Revision and response profile (skepticism, revision willingness, detail preference, noise)	Controls how new text changes prior beliefs and how answers are emitted
e_i	Episodic traces of recently read teaching events	Supports short-term retrieval even when durable KC mastery is still weak
$\hat{m}_{i,k}$	Tutor-visible BKT mastery estimate	Drives reading adaptation but does not directly change hidden learning

Table 1: Learner-state variables.

as easier to encode than longer ones. *Refutation strength* approximates how explicitly the event confronts a likely misconception; in the current implementation it is triggered by lexical cues such as misconception, incorrectly, rather than, instead of, or do not, and it scales misconception reduction during learning. *Refresh* marks lower-gain reinforcement events when review KCs are included in a passage; operationally, refresh is a lightweight heuristic rather than a discourse parser.

This representation keeps learner memory close to the CI notion of a textbase while remaining simple enough to generate, score, and inspect at scale. Encoded events create episodic traces that preserve the event text, KC links, clarity, refutation strength, readability score, and later distortion, making subsequent answer behavior inspectable. When provenance is needed, it is retained at the reading-asset level through source chunk identifiers rather than per-event source spans.

For example, the sentence pair “Plants do not get food from soil. They make food through photosynthesis.” is stored as a proposition-like teaching event linked to a photosynthesis KC:

```
{
  proposition_id: "prop_001",
  text: "Plants do not get food from soil.
  They make food through photosynthesis.",
  kc_ids: {"KC_photosynthesis_food_source"},
  clarity: 0.92,
  refutation_strength: 1.00,
  is_refresh: false
}
```

The record is operational rather than a full semantic proposition.

4.4 Comprehension as textbase construction and situation-model integration

The reading update proceeds in two stages. First, the learner encodes teaching events into a textbase-like episodic memory. Second, the learner integrates those events with prior knowledge to update correct knowledge and misconception weights. Operationally, the encoding strength of event p for learner i reading passage j is modeled as

$$\text{enc}_{i,p} = \sigma(\alpha + 1.1 \text{clarity}_p + 0.55 \text{match}_{ij} + \beta^\top r_i + 0.3(1 - \bar{m}_{i,p}) + 0.35 \text{attention}_i). \quad (3)$$

Here, $\bar{m}_{i,p}$ is mean prior mastery over the event’s target KCs and $\beta^\top r_i$ collects vocabulary, inference, strategy, and background-knowledge effects. If an event is encoded, it creates an episodic trace whose later activation depends on recency, rehearsal, lexical overlap with the question, and distortion noise.

Once encoded, an event updates KC state by increasing correct knowledge and reducing misconception weight. In the implementation, correct-knowledge gain is stronger for clearer events, better reader-text matches, and learners with higher revision willingness and detail preference. Misconception reduction is scaled by the event’s refutation strength, consistent with KREC’s emphasis on how prior incorrect knowledge is confronted during reading (Kendeou and O’Brien, 2014; Kendeou, 2024). Refresh events receive a smaller gain multiplier than new target events.

The coefficients in Equation 3 and Equation 4 were selected during simulator development and

then held fixed across subject comparisons. They encode qualitative assumptions rather than fitted human-data parameters: clarity, match, relevant prior knowledge, attention, and strategy use increase encoding; misconception strength and item difficulty reduce correct responding; and retrieved traces provide partial support.

4.5 Score-based answer option selection from learner memory

At assessment time, the learner does *not* receive the original passage. Instead, the simulator retrieves relevant traces from memory and computes a score for each item from hidden mastery, misconception weight, retrieved-trace support, and response-style variables. The guiding principle is still epistemic boundedness in the sense of [Yuan et al. \(2026\)](#): the learner answers only from what it has encoded and retained. However, the current implementation uses score-based option selection rather than free-form generative reasoning.

For item q , let $\bar{m}_{iq}$ be mean correct knowledge over the target KCs, $\bar{w}_{iq}$ the mean misconception weight, and τ_{iq} the mean activation of retrieved traces. The item’s response utility is

$$u_{iq} = -b_q + 2.2\bar{m}_{iq} + 0.38\tau_{iq} - 1.25\bar{w}_{iq} + 0.35\text{attention}_i - 0.25\text{guess}_i, \quad (4)$$

where b_q is a difficulty offset derived from the item’s difficulty band. The probability of a correct answer is then

$$P(y_{iq} = 1) = \sigma(u_{iq}), \quad (5)$$

clipped to a small interior interval to avoid degenerate probabilities.

If the learner answers incorrectly, the selected distractor is *not* sampled uniformly. Instead, distractors are scored by lexical overlap between the retrieved trace text and the distractor text/rationale, plus learner-specific guessing bias. This makes misconception-consistent wrong options more likely than random errors when the learner’s memory traces support them. The simulator also records an explicit epistemic-state summary for each response, including target KCs, correct-knowledge support, misconception support, and the IDs of the retrieved traces used during scoring.

Subject	Ch.	LOs	KCs
Computer Science	16	53	131
General Biology	20	60	172
Inorganic Chemistry	12	57	177

Table 2: Ontology sizes shown in the Ontology Atlas coverage view for the three ontologies used in the experiments.

5 Adaptive Personalization with Bayesian Knowledge Tracing

The tutor maintains a separate BKT model for each KC. After each question response, the tutor updates the KC-specific mastery estimate $\hat{m}_{i,k}$ using standard BKT parameters for initial mastery, learning, guess, and slip ([Corbett and Anderson, 1994](#)). These estimates drive reading adaptation but do not directly modify hidden learner state.

We use BKT as the tutor-visible observation model, not as the cognitive model of reading. Passages update hidden semantic state; quiz responses provide noisy evidence for KC mastery estimates. BKT is useful here because it gives the adaptation policy an interpretable KC-level state, so we report both hidden-state summaries and tutor-side BKT gain.

The adaptation policy maps low estimated mastery to more supportive readings. In the adaptive condition, low $\hat{m}_{i,k}$ can increase explanatory depth, example density, explicit misconception refutation, repeated weak-KC coverage, and reader-text matching. Higher $\hat{m}_{i,k}$ yields leaner variants. The control condition uses the same ontology, source corpus, and assessment machinery, but holds the reading configuration fixed rather than conditioning it on $\hat{m}_{i,k}$.

6 Experimental Setup

We ran fixed-iteration experiments in three subject ontologies: foundations of computer science, general biology, and introductory inorganic chemistry. [Table 2](#) summarizes the ontology sizes surfaced in Ontology Atlas for those three subjects.

For each subject, we sampled four LOs across two chapters, created 50 matched simulated learners per condition, ran three reading-assessment cycles per LO, and generated three KC-aligned multiple-choice items per cycle. This yields 1,800 item-level responses per condition per subject.

Adaptive and control runs were matched at initialization: each control run reused the same learner

seeds and initial hidden states as its adaptive counterpart. Assessment delivery context was set to summative, so quiz answers updated tutor-visible BKT state but did not directly change hidden semantic knowledge. We report observed accuracy, tutor-side BKT gain, hidden mastery gain, and misconception reduction. Because learner-level paired trajectories are preserved only for accuracy and BKT gain, inferential tests are limited to those outcomes.

7 Results

7.1 Subject-level outcomes

Figure 4 and Table 3 summarize adaptive-minus-control deltas. Computer science shows the clearest benefit: accuracy improved from 0.834 to 0.865 ($\Delta = +0.031$, 95% CI [$+0.004$, $+0.058$], $p = 0.026$), and learner-level BKT gain improved from 0.416 to 0.454 ($\Delta = +0.038$, 95% CI [$+0.002$, $+0.074$], $p = 0.037$). Chemistry was positive but inconclusive (accuracy $\Delta = +0.019$, $p = 0.104$; BKT gain $\Delta = +0.009$, $p = 0.540$). Biology was neutral to slightly negative (accuracy $\Delta = -0.005$, $p = 0.691$; BKT gain $\Delta = -0.014$, $p = 0.468$). Descriptive hidden-state endpoints in Table 4 favored adaptation for hidden mastery in all three subjects and for misconception reduction in computer science and chemistry.

Subject	Cycle	Observed Accuracy			Tutor-side BKT gain		
		Adaptive	Control	Δ	Adaptive	Control	Δ
Computer Science	1	0.802	0.760	+0.042	0.180	0.157	+0.024
	2	0.890	0.853	+0.038	0.168	0.146	+0.023
	3	0.904	0.890	+0.014	0.106	0.114	-0.008
General Biology	1	0.794	0.795	-0.001	0.174	0.167	+0.007
	2	0.864	0.849	+0.015	0.122	0.111	+0.011
	3	0.877	0.906	-0.029	0.073	0.106	-0.033
Inorganic Chemistry	1	0.807	0.797	+0.010	0.190	0.190	+0.000
	2	0.868	0.828	+0.040	0.146	0.136	+0.010
	3	0.885	0.878	+0.007	0.092	0.092	+0.000

Table 3: Subject-level adaptive versus control outcomes.

7.2 Cycle-level trajectories

Figure 5 shows that adaptive reading led control in every computer-science and chemistry cycle, with the largest chemistry margin in cycle 2. Biology showed a crossover pattern: nearly tied in cycle 1, adaptive ahead in cycle 2, and control ahead in cycle 3.

8 Discussion

The results suggest domain-dependent effects of the adaptive policy. Computer science provides the most consistent evidence of benefit, chemistry

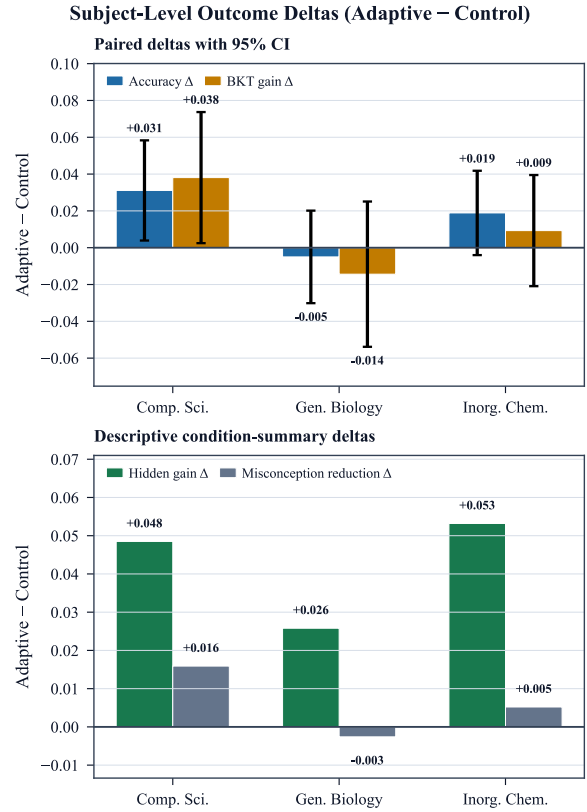


Figure 4: Subject-level adaptive-minus-control deltas.

Subject	Hidden mastery gain			Misconception reduction		
	Adapt.	Ctrl.	Δ	Adapt.	Ctrl.	Δ
Computer Science	0.521	0.473	+0.049	0.072	0.056	+0.016
General Biology	0.388	0.362	+0.026	0.045	0.048	-0.003
Inorganic Chemistry	0.509	0.456	+0.053	0.061	0.056	+0.005

Table 4: Descriptive hidden-state outcomes from saved experiment summaries.

shows smaller positive but inconclusive gains, and biology shows a mixed pattern in which hidden mastery improved descriptively without improving observed accuracy or tutor-side BKT gain.

This divergence between hidden and observed outcomes is informative. In the simulator, answers depend on hidden KC knowledge, episodic traces, misconception weight, item difficulty, attention stability, and distractor overlap. A reading can therefore improve hidden knowledge while producing little observed gain if retrieval is brittle, items are noisy, or distractors remain competitive.

The results also illustrate the role of Ontology Atlas in auditing ontology granularity and coverage. Adaptive benefit varied by LO and subject, so ontology quality and KC granularity are not merely upstream bookkeeping issues. If an LO is too broad, if its KCs are diffuse, or if retrieved

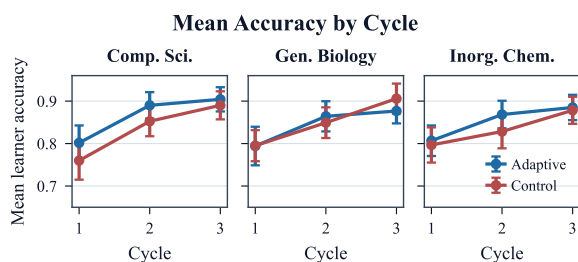


Figure 5: Mean learner-level quiz accuracy by fixed iteration cycle.

evidence is weakly aligned with target KCs, both personalization and evaluation become harder to interpret.

9 Conclusion

We presented a full-stack offline framework for evaluating adaptive personalization of educational readings with theory-grounded simulated learners. The framework links ontology-grounded content generation, KC-level BKT mastery tracking, Ontology Atlas ontology authoring, CI-based comprehension, DIME-style reader heterogeneity, KREC-style misconception revision, and score-based answer selection over explicit learner memory. Across three sampled subject ontologies, adaptive reading yielded reliable gains in computer science, smaller inconclusive gains in chemistry, and neutral to slightly negative observed outcomes in biology.

Limitations

The framework has several limitations. First, simulated learners are not substitutes for real students. The simulator is designed for inspectable offline policy screening, not as a validated estimator of classroom effect sizes, and it has not yet been calibrated against human response distributions, learning curves, distractor choices, misconception persistence, or subgroup heterogeneity.

Second, the modeled instructional setting is intentionally narrow: learners read independently, assessments provide observations but no direct instruction, and peer interaction, instructor intervention, classroom discussion, feedback, and retrieval-practice effects are outside the simulation. This isolates reading as the intervention, but it does not approximate courses where instructors diagnose errors, peers explain concepts, or quizzes teach through feedback and retrieval practice.

Third, the ontology and labeling pipeline are

central sources of uncertainty. The ontologies are corpus-grounded and built from the same Wikibooks textbooks used for retrieval and generation. Errors in LO/KC granularity, chunk labels, retrieved evidence, or item alignment can propagate through generation, adaptation, and evaluation, so cross-domain differences may reflect content-pipeline quality as well as the adaptive policy. Porting to a new course therefore still requires domain review of the ontology, labels, generated readings, and assessment items rather than only rerunning the automatic pipeline.

Fourth, the simulator uses simplified operational heuristics: clarity is approximated from length, refutation strength from lexical cues, and the encoding and response equations use fixed coefficients. Sensitivity to coefficient perturbations, alternative detectors, and ablations remains untested.

Fifth, the readability signal is deliberately shallow. New Dale–Chall provides an open, reproducible proxy based on word familiarity and sentence length (Chall and Dale, 1995). It was used because the learner model includes vocabulary and background-knowledge factors: Flesch Reading Ease is driven by average sentence length and average syllables per word (Flesch, 1948), whereas New Dale–Chall incorporates word familiarity in addition to sentence length. This makes New Dale–Chall a closer operational proxy for whether a passage contains vocabulary likely to be unfamiliar to a reader, including short domain terms that syllable-based formulas may not penalize. However, it does not model discourse cohesion, diagrams, mathematical notation, domain-specific prior vocabulary, or whether a learner already knows terms such as *ATP*, *loop*, or *ion*. Future work should compare New Dale–Chall with Flesch–Kincaid, domain-vocabulary measures, and learned reader–text difficulty models.

Sixth, score-based answer selection supports bounded competence and auditability, but it may under-represent open-ended reasoning, explanation quality, and strategic test-taking. We also do not compare against constrained LLM-based student simulators or other cognitive simulators, and pre-processing artifacts may depend on the embedding and instruction models used.

Finally, the saved result format preserves learner-level paired trajectories for accuracy and BKT gain, but only condition-level summaries for hidden mastery and misconception endpoints. Future runs should preserve learner-level hidden-state trajec-

ries for paired analysis.

Ethics Statement

The system uses open educational content and simulated learners. Any deployment decisions affecting students require human-subject validation under appropriate institutional review and privacy safeguards.

References

- Shivam Bansal and Chaitanya Aggarwal. 2026. [textstat: Calculate statistical features from text](#). Python package, version 0.7.13. Accessed 2026-03-30.
- Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, Cambridge, MA.
- Albert T. Corbett and John R. Anderson. 1994. [Knowledge tracing: Modeling the acquisition of procedural knowledge](#). *User Modeling and User-Adapted Interaction*, 4(4):253–278.
- Jennifer G. Cromley and Roger Azevedo. 2007. [Testing and refining the direct and inferential mediation model of reading comprehension](#). *Journal of Educational Psychology*, 99(2):311–325.
- Jennifer G. Cromley, Lindsey E. Snyder-Hogan, and Ulana A. Luciw-Dubas. 2010. [Reading comprehension of scientific text: A domain-specific test of the direct and inferential mediation model of reading comprehension](#). *Journal of Educational Psychology*, 102(3):687–700.
- Xinyi Ding and Eric C. Larson. 2021. [On the interpretability of deep learning based models for knowledge tracing](#). *Preprint*, arXiv:2101.11335.
- Rudolf Flesch. 1948. [A new readability yardstick](#). *Journal of Applied Psychology*, 32(3):221–233.
- Mark J. Gierl, Okan Bulut, Qi Guo, and Xinxin Zhang. 2017. [Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review](#). *Review of Educational Research*, 87(6):1082–1116.
- Xiaoli Huang, Wei Xu, and Ruijia Liu. 2025. [Effects of intelligent tutoring systems on educational outcomes: Evidence from a comprehensive analysis](#). *International Journal of Distance Education Technologies*, 23(1):1–25.
- Tanja Käser and Giora Alexandron. 2024. [Simulated learners in educational technology: A systematic literature review and a turing-like test](#). *International Journal of Artificial Intelligence in Education*, 34(2):545–585.
- Panayiota Kendeou. 2024. [A theory of knowledge revision: The development of the krec framework](#). *Educational Psychology Review*, 36(2):44.
- Panayiota Kendeou and Edward J. O’Brien. 2014. The knowledge revision components (krec) framework: Processes and mechanisms. In David N. Rapp and Jason L. G. Braasch, editors, *Processing Inaccurate Information: Theoretical and Applied Perspectives from Cognitive Science and the Educational Sciences*, pages 353–377. MIT Press, Cambridge, MA.
- Walter Kintsch. 1988. [The role of knowledge in discourse comprehension: A construction-integration model](#). *Psychological Review*, 95(2):163–182.
- Benoît Lemaire, Guy Denhière, Cédric Bellissens, and Sandra Jhean-Larose. 2006. [A computational model for simulating text comprehension](#). *Behavior Research Methods*, 38(4):628–637.
- Angélique Létourneau, Marion Deslandes Martineau, Patrick Charland, John Alexander Karran, Jared Boasen, and Pierre-Majorique Léger. 2025. [A systematic review of AI-driven intelligent tutoring systems \(ITS\) in K-12 education](#). *npj Science of Learning*, 10(1):29.
- Paul van den Broek and Panayiota Kendeou. 2008. [Cognitive processes in comprehension of science texts: The role of co-activation in confronting misconceptions](#). *Applied Cognitive Psychology*, 22(3):335–351.
- Kurt VanLehn, Stellan Ohlsson, and Rod Nason. 1994. Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education*, 5(2):135–175.
- Wikibooks contributors. 2026a. Foundations of Computer Science. https://en.wikibooks.org/wiki/Foundations_of_Computer_Science. Accessed 2026-05-13.
- Wikibooks contributors. 2026b. General Biology. https://en.wikibooks.org/wiki/General_Biology. Accessed 2026-05-13.
- Wikibooks contributors. 2026c. Introduction to Inorganic Chemistry. https://en.wikibooks.org/wiki/Introduction_to_Inorganic_Chemistry. Accessed 2026-05-13.
- Zhihao Yuan, Yunze Xiao, Ming Li, Weihao Xuan, Richard Tong, Mona T. Diab, and Tom Mitchell. 2026. [Towards valid student simulation with large language models](#). *Preprint*, arXiv:2601.05473.

Toward Cross-Domain Automated Feedback: A Comparative Evaluation of Open-Source Models across Diverse Student Assessment Types

Muhammad Haseeb¹, Min Paing Hmue¹, Ahmad Imam Amjad²,
Maaz Amjad¹, Victor S. Sheng¹

¹Texas Tech University, Texas, USA, ²The University of Punjab, Pakistan,
mhaseeb@ttu.edu, mhmue@ttu.edu, ahmadimamamjad@gmail.com, maaz.amjad@ttu.edu, victor.sheng@ttu.edu

Abstract

Constructive, personalized, and timely feedback is essential to student learning. However, providing such feedback in large classes remains a major challenge. Large language models (LLMs) offer alternatives to support instructors by generating relevant feedback that highlights both student strengths and areas for improvement. Nevertheless, most existing LLM-based feedback systems rely on proprietary APIs and are often tailored to specific tasks, which limits their accessibility, flexibility, and applicability in resource-constrained educational settings. In this study, we investigate the potential of two open-source LLMs (DeepSeek R1 and Qwen 3.5) to support automated feedback generation across three disciplines (e.g., programming assignments, essays, and mathematics problems). We evaluate two prompting strategies (unified and multi-agent) across these domains and use human judgment criteria to assess feedback quality. Through this analysis, we examine the potential of open-source models as practical, scalable alternatives to proprietary API-based systems for developing freely accessible feedback-generation tools. Our results show that a single open-source model can generate useful feedback across diverse domains, though with varying effectiveness. DeepSeek R1 performs better on reasoning-intensive tasks such as mathematics, while Qwen 3.5 works best for holistic tasks such as writing, but both models struggle with programming tasks. We find that model architecture matters more than prompting strategy, and reasoning-optimized models excel in structured domains, while general-purpose models perform better on holistic tasks. Finally, we conclude that a multi-agent approach does not consistently guarantee improved results over a single unified LLM approach.

1 Introduction

Feedback plays an important role in the learning process (Hattie and Timperley, 2007). Constructive

and timely feedback helps students identify gaps in their understanding, refine their skills, and improve academic performance (Cavalcanti et al., 2021). However, the usefulness of feedback in educational settings depends not only on its quality but also on how quickly the feedback is given. Students report significantly lower motivation when feedback is delayed and takes more than 10 days to arrive (Fisher et al., 2025). In contrast, personalized and timely feedback can improve academic performance and overall learning outcomes (Aucejo and Wong, 2025). However, instructors find it difficult to provide timely, high-quality, and individualized feedback in large classes and resource-constrained educational settings due to workload constraints (Messer et al., 2024). This leads to delayed feedback, inconsistencies in grading, poor feedback quality, and tracking individual student progress over time becomes more challenging (Haseeb et al., 2026). To address these challenges, artificial intelligence (AI) and large language models (LLMs) offer opportunities to complement instructor support by enabling automated feedback generation and addressing scalability issues.

LLM-based systems have been proposed to automate the process of feedback generation (Murtazin et al., 2025; Haseeb et al., 2026). These systems help reduce feedback turnaround time from days to seconds (Haseeb et al., 2026), and mitigate delays associated with manual grading and feedback delivery (Dong et al., 2024; Guizani et al., 2025; Kinder et al., 2025; Nkoyo et al., 2025). Recent studies demonstrate that LLM-generated feedback can support students' learning goals, enhance motivation, and improve engagement, especially when the feedback is adaptive and responsive to learners' needs (Murtazin et al., 2025; Kinder et al., 2025), which makes them well-suited for both traditional classroom and online learning environments (Cisneros-González et al., 2025).

Most state-of-the-art feedback generation sys-

tems used in AI-assisted educational applications rely on proprietary models (e.g., GPT-4, Gemini, and Claude). However, dependence on these models creates transparency, reproducibility, and accessibility-related issues. These limitations are important because they hinder the practical deployment of such tools in resource-constrained academic institutions, where reliance on commercial AI systems or the computational infrastructure required to deploy large-scale models may not be practical or sustainable. Therefore, it is important to explore whether open-source models can serve as a reasonable alternative that addresses these limitations for automated feedback generation.

Open-source language models provide freely accessible alternatives for educators and practitioners. In educational settings, they are particularly valuable because they can support transparency, reproducibility, adaptability, local deployment, and cost control (Lin et al., 2026). In addition, open-source models have shown promising results in targeted feedback-generation tasks. For example, open-source models have been found to perform nearly on par with proprietary models in generating and evaluating feedback in programming education (Koutchme et al., 2025). However, prior research has largely examined these models in isolated educational tasks and within narrow disciplinary contexts. Most of the existing literature is concentrated on a single-domain setup, which includes computer science-related settings (e.g., programming education (Koutchme et al., 2025; Cisneros-González et al., 2025)), writing assessment (Alnemrat et al., 2025; Gilmore, 2025), and mathematical reasoning tasks (Lee et al., 2026; Rigaud Téllez et al., 2024). However, relying on separate tools for different assessment types is often impractical in real educational settings. Consequently, it remains unclear whether a single open-source model can generate useful feedback across diverse assessment types and disciplines (e.g., programming, essay writing, and mathematics). This gap motivates the need for a broader cross-disciplinary evaluation of open-source language models for feedback generation.

To the best of our knowledge, no previous study has examined a single LLM-based model (as a unified framework) for feedback generation across three disciplines: (i) mathematics, (ii) academic writing, and (iii) programming. To address this gap, we extend previous work on LLM-based feedback generation (Haseeb et al., 2026) to examine LLM-based feedback generation across mathemat-

ics, academic writing, and programming within a single unified framework.

Research Question: To what extent can a single task-specialized open-source model generate useful and pedagogically meaningful feedback across different assessment types while maintaining consistent feedback quality? For example, can a coding-specialized model also provide useful feedback on essay-based assignments and mathematics problems? We further hypothesize that decomposing feedback generation into a multi-agent setup can yield deeper and more domain-specific feedback insights than a unified single prompt approach.

The main contributions of this paper are as follows:

1. We extend LLM-based feedback generation beyond programming to two additional disciplines: (i) mathematics, and (ii) academic writing.
2. We collect and curate datasets for each of the three disciplines, and then examine 2 open-source models across 3 domains, to assess feedback quality.

2 Methodology

We evaluate LLM-based feedback generation using two prompting approaches applied to two open-source models across three educational domains: (i) programming, (ii) academic writing, and (iii) mathematics. Figure 1 illustrates the overall framework. In the **Unified Approach**, each model independently receives the full context, including the problem, student submission, and evaluation criteria, in a single prompt and produces feedback in one inference call.

In the **Multi-Agent Approach**, feedback generation is decomposed into specialized subtasks assigned to distinct agents, each focused on a specific aspect of evaluation, followed by a synthesis step where applicable. Both approaches are evaluated across programming, academic writing, and mathematics to assess whether prompting architecture affects feedback quality across domains.

2.1 Datasets

To evaluate LLM feedback generation across educational domains, we curate datasets from three distinct disciplines: programming, academic writing, and mathematics. Each dataset contains student

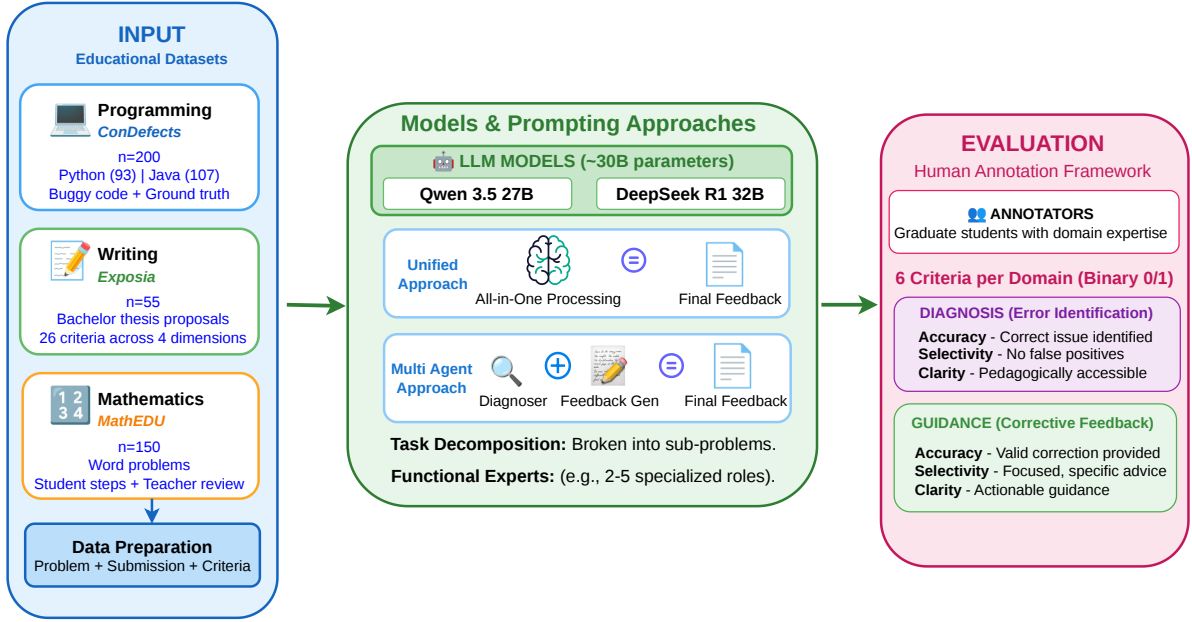


Figure 1: Framework for LLM-based feedback generation across domains with unified and multi-agent prompting.

submissions with errors, along with their associated ground truth. This enables systematic evaluation of feedback quality. Table 1 summarizes the key characteristics of each dataset.

Dataset	Samples	Languages	Source
ConDefects	200	Python (93) Java (107)	(Wu et al., 2023)
Exposia	55	English	(Zyska et al., 2026)
MathEDU	150	English	(Hsu et al., 2026)

Table 1: Datasets Overview

2.1.1 Programming: ConDefects

We use ConDefects dataset (Wu et al., 2023) for evaluating the feedback in the programming task. It consists of real buggy code submissions from competitive programming contests on AtCoder. This dataset contains student errors made during the competition, which reflect students’ understanding of the code, misconceptions, and mistakes.

We use 200 students’ submissions across two programming languages: 93 submissions written in Python, and 107 submissions in Java. Each sample includes four elements: (i) a problem description, (ii) the student’s buggy code, (iii) the corrected version, and (iv) the exact line where the student made a mistake. These problems are of intermediate difficulty level, and each student submission in this set exhibits distinct programming concepts, error types, and logical errors. Since we did not have access to the reference feedback, we rely on human

assessment to evaluate the quality of LLM feedback. The evaluation assesses whether the LLM accurately identifies the bug while providing pedagogically useful and personalized guidance.

2.1.2 Writing: Exposia

For the academic writing domain, we use the Exposia dataset (Zyska et al., 2026), which comprises 55 bachelor’s thesis research proposals from TU Darmstadt. Each proposal represents a student’s initial draft and the final proposal after incorporating the teacher’s and fellow students’ feedback. Each proposal includes a problem statement, a literature review, a methodology, and a writing quality assessment.

The dataset also includes a well-defined scoring criterion that teachers and fellow student reviewers use to assess submission quality. For our evaluation, we use teacher feedback and assigned scores as references to guide human evaluators, but we do not treat them as absolute ground truth, since assessing academic writing quality is inherently subjective. Therefore, human evaluators utilize their own judgment to make final quality assessments, informed by but not bound to instructor scores.

We utilize the 26 criteria across four dimensions provided with the dataset (Zyska et al., 2026), originally used by teachers to assess submissions and generate feedback. We incorporate these criteria directly into our prompts to guide LLM-based feedback generation.

Problem & Research Question (MO1-MO7, 9 points): Evaluates if there is an engaging hook, a clear problem statement, a domain specification, a relevant problem, a discussion of current approaches, a properly formulated research question, and scope feasibility.

Literature & Theory (AP1-AP7, BI1-BI2, 13 points): Evaluates state-of-the-art coverage, relevance demonstration, weakness identification, differentiation from prior work, synthesis quality, and theoretical framework quality.

Methodology (AP8, MD1-MD7, 8 points): Examines methodology availability, completeness, relevance, reference to existing materials, discussion of difficulties, limitations acknowledgment, and implementation details.

Writing Quality (LA1-LA2, 4 points): Evaluates the language quality (grammar, clarity, terminology) and structural coherence (logical flow, consistent language, and common thread).

Each criterion uses a defined rubric with scores ranging from 0-1 or 0-2 points, yielding a maximum of 33 points. We use the expert annotations (type: “draft”, role: “expert”) as ground truth for evaluating LLM scoring accuracy.

2.1.3 Mathematics: MathEDU

For the mathematics domain, we use 150 problems from the MathEDU dataset (Hsu et al., 2026), which is designed to evaluate mathematical reasoning and tutoring capabilities. Each sample consists of a word problem, the student’s incorrect answer, and their solution process, which shows the students’ reasoning steps. The dataset contains diverse mathematical concepts, such as ratio calculations, arithmetic operations, distance-time-speed problems, probability, permutations, and geometric reasoning. Each sample includes a *teacher_review* field containing expert feedback on the student’s error, and provides a reference ground truth for evaluating the quality of LLM-generated feedback.

2.2 Open-source Models

We select two state-of-the-art open-source large language models, each model representing a distinct architectural paradigm: (i) a general-purpose model, and (ii) a reasoning-oriented model. Both models operate at a comparable scale (approx 30B parameters). This ensures that any observed performance differences reflect model design and reasoning capabilities rather than the parameter count.

1. **Qwen 3.5 27B** (Yang et al., 2025) is a general-purpose model achieving state-of-the-art performance on standard benchmarks. We use the BF16 precision variant with thinking mode disabled via Ollama, generating responses directly without explicit reasoning traces. This setup reflects standard deployment in efficiency-constrained environments for educational feedback.
2. **DeepSeek R1 32B** (Guo et al., 2025) is a reasoning-distilled model. It compresses chain-of-thought capabilities from larger models into a smaller architecture. The model generates explicit thinking traces before producing the final response. We use the FP16 precision variant with default settings.

Our model selection balances three considerations. First, comparable parameters ensure a fair comparison. Second, both models are top performers within their paradigm at this scale. Third, the 30B parameter range allows deployment on a single-GPU infrastructure. This reflects practical deployment constraints, supports replicability, and improves accessibility in low-resource educational environments. This pairing helps examine whether explicit model reasoning improves feedback quality or whether general-purpose generation that follows the rubrics suffices to maintain feedback quality in educational domains.

2.3 Prompting Approaches

We evaluate two distinct prompting approaches for generating educational feedback in three disciplines (programming, essay writing, and mathematics): (i) a unified single-prompt approach, and (ii) a multi-agent prompting approach. These approaches are compared in terms of the quality of feedback generated between both models.

2.3.1 Unified Approach

The unified approach generates feedback through a single LLM call. The model receives the full context problem, student submission, and evaluation criteria, and produces a single feedback response.

For the programming task (see Figure 2), the prompt includes the problem description, related test cases, the programming language, and the student’s buggy code. It also includes instructions to identify the error and provide feedback on a conceptual misunderstanding. The output is 150-250

words, which focuses on bug identification and pedagogical guidance. For the writing task (see Figure 3), the prompt includes the research proposal and 26 scoring criteria organized across four dimensions. The model produces scores in a standardized format. It then gives 150-250 words of feedback to highlight key issues and suggest improvements. For the mathematics task (see Figure 4), the prompt includes the problem, the student’s answer, and their solution process. The model identifies incorrect steps, diagnoses underlying misconceptions, and provides guidance to address these gaps. This approach mirrors how educators review student work and assess all aspects of their work at once. Moreover, this approach is computationally efficient, requiring only one inference call per student submission.

2.3.2 Multi-Agent Approach

The multi-agent approach divides feedback generation into specialized sub-tasks. Each agent is assigned a specific evaluation or feedback generation step, and, if necessary, a synthesis step to produce coherent and comprehensive feedback. Each agent is implemented as a separate LLM call, where the output of one stage serves as input to the next.

In the programming domain (see Figure 5–6), two agents are utilized:

- **Error Analyzer:** This agent produces a structured diagnosis that includes the location of the error, the type of error, the root cause, and the misconception of the student and the direction of the correction.
- **Feedback Generator:** This agent receives the diagnosis, the problem statement, and the submission, and crafts pedagogically helpful feedback for the student.

In the writing domain, four dimension-specific agents are employed in conjunction with a synthesizer:

- **Problem and RQ Agent:** Scores the problem and research question and provides 2-3 sentence feedback on clarity and relevance.
- **Literature Agent:** Evaluates the quality of the literature review section and provides 2-3 sentences on quality, coverage, and synthesis.
- **Methodology Agent:** Evaluates completeness and relevance of methodology and provides feedback on quality, completeness, and rigor.

- **Writing Quality Agent:** Scores language quality, consistency, and structure while providing feedback.
- **Synthesizer:** Integrates all scores and analyses to formulate a unified feedback response of 150 to 250 words for the student.

In the mathematics domain, two agents are utilized:

- **Error Analyzer:** Outputs error location along with error type and conceptual gap, with a small description of the mistake.
- **Feedback Generator:** Receives diagnosis with problem description and submission, and formulates personalized feedback for students in 50-120 words.

We include representative prompts for the programming domain in the appendix (Figures 2, 5, 6). Prompts for writing and mathematics follow the same structure and are publicly available in the accompanying repository, along with sample model outputs for all domains¹.

2.3.3 Design Rationale

Our architectures are motivated by domain-specific task-decomposition principles from prior work, in which complex evaluation tasks are decomposed into distinct role-based cognitive sub-tasks (Wang et al., 2025). This persona-based decomposition improves the analytical depth and feedback quality.

Programming. We adopt a two-stage pipeline (diagnosis → feedback) following code review frameworks (Yang et al., 2024). Debugging in programming first requires identifying the precise source, location, and the type of error, whereas the corrective guidance needs to provide pedagogically meaningful feedback. Separating bug identification helps the respective agent to focus more on the required task. This decomposition addresses the limitations of LLMs in generating correct code without accurately localizing conceptual errors.

Writing. We decompose our evaluation strategy by rubric dimensions, which aligns with writing assessment practices where content, structure, and language require distinct evaluative expertise (Hashemi et al., 2024). The four-agent structure maps directly to our selected dataset rubric dimensions. Finally, a synthesizer integrates these

¹<https://github.com/haseeb2001/crossdomainfeedback>

dimension-specific analyses into coherent feedback, resolving potential conflicts and avoiding fragmented responses.

Mathematics. Our pipeline mirrors cognitive tutoring approaches, in which diagnosing misconceptions precedes corrective instruction (Liu et al., 2023). In mathematical problems, errors are often rooted in conceptual misunderstandings or omissions during the solving process, rather than surface-level mistakes; therefore, step-by-step error diagnosis enables targeted, personalized remediation rather than generic solution guidance.

The multi-agent paradigm draws from recent work demonstrating that task decomposition improves LLM performance on complex reasoning tasks (Wang et al., 2025). We hypothesize that specialized agents can provide deeper, domain-specific, and more precise feedback than a unified approach, especially for tasks like mathematics that require pedagogical reasoning.

2.4 Experimental Setup

All experiments were conducted on NVIDIA H100 GPUs (96GB VRAM) using Ollama (v0.6.8) for local inference. Qwen 3.5 27B was executed on a single GPU, whereas DeepSeek R1 32B required two GPUs due to higher memory requirements. For both models, we used deterministic decoding with a temperature set to 0 and a maximum output length of 4096 tokens. All other generation parameters were kept at their default values. These settings were applied consistently across all experiments. All our prompts and inference scripts are publicly available ².

2.5 Evaluation Framework

We evaluate generated feedback through human annotation using six criteria per domain, organized into two dimensions: *diagnosis* (error identification) and *guidance* (corrective feedback).

Each dimension is assessed along three aspects: **Accuracy**—correctly identifies the underlying issue and/or provides a valid correction; **Selectivity**—avoids false positives and focuses only on relevant issues; **Clarity**—pedagogically accessible and appropriate for the student’s level.

For programming, we adopt evaluation criteria from (Koutchene et al., 2025). For writing and mathematics, we developed parallel criteria to enable consistent cross-domain comparison under a

unified evaluation framework. The full set of evaluation criteria is provided in Table 2.

Domain	Code	Definition
<i>Programming (Explanation / Fix)</i>		
	EA	Bug description matches ground truth
	ES	No false positives identified
	EC	Explanation understandable to a novice
	FA	Fix would resolve the bug
	FS	No extraneous modifications
	FC	Clear and usable code suggestions
<i>Writing (Issue / Suggestion)</i>		
	WA	Writing issue correctly identified
	WS	No overcriticism of acceptable choices
	WC	Understandable to student writer
	GA	Suggestion would improve writing
	GS	Focused and specific advice
	GC	Actionable with concrete guidance
<i>Mathematics (Error / Solution)</i>		
	MA	Mathematical error correctly identified
	MS	Valid alternative paths not penalized
	MC	Appropriate for student level
	SA	Guidance leads to correct solution
	SS	Addresses specific error only
	SC	Step-by-step actionable guidance

Table 2: Evaluation criteria across domains. All criteria scored binary (0/1).

Annotation Protocol. Evaluation was conducted by four annotators for programming and mathematics, and three annotators for writing, proportional to dataset size. All annotators were graduate students with relevant domain expertise. Each criterion was scored binary (0 = No, 1 = Yes). Final scores were aggregated by averaging across annotators for each sample and then across all samples. Inter-annotator agreement was measured using Cohen’s κ on overlapping samples, which indicates moderate agreement ($\kappa = 0.48$) across configurations.

3 Results

This section presents evaluation results across three educational domains: programming (Section 3.1), writing (Section 3.2), and mathematics (Section 3.3). All results are reported using the six-criterion evaluation framework introduced in Section 2.5 (see Table 2 for criterion definitions).

3.1 Programming Results

Programming feedback presents the greatest challenge across all three domains, with average scores (0.27-0.48), which is substantially lower than mathematics (0.70-0.91) and writing (0.35-0.81). Table 3 presents the complete results.

²<https://github.com/haseeb2001/crossdomainfeedback>

DeepSeek R1 outperforms Qwen 3.5 across all conditions (0.43-0.48 vs 0.27-0.29), with the unified approach yielding marginally better results than the multi-agent approach for both models. A striking pattern emerges in criterion-level scores: explanation clarity (EC) approaches perfection (0.82-1.00) across all configurations, whereas explanation accuracy (EA: 0.12-0.20) and fix accuracy (FA: 0.09-0.55) remain comparatively low. This difference shows that models can produce fluent, well-articulated feedback but often fail to correctly identify underlying bugs and conceptual errors, rendering their feedback ineffective. This pattern of fluent feedback that fails to identify actual bugs confirms prior observations on programming feedback (Haseeb et al., 2026).

Configuration	EA	ES	EC	FA	FS	FC	Avg
Qwen 3.5-U	0.17	0.10	1.00	0.14	0.15	0.20	0.29
Qwen 3.5-MA	0.12	0.06	0.97	0.09	0.13	0.27	0.27
DeepSeek R1-U	0.20	0.18	0.86	0.55	0.55	0.52	0.48
DeepSeek R1-MA	0.14	0.15	0.82	0.54	0.54	0.40	0.43

Table 3: Programming domain results (n=200). U = Unified, MA = Multi-Agent.

3.2 Writing Results

Qwen 3.5 substantially outperforms DeepSeek R1 across all conditions, achieving an average score of 0.81 compared to 0.35–0.38 for DeepSeek. This performance gap is more pronounced in accuracy metrics: Qwen achieves 0.91–0.93 on both issue identification (WA) and suggestion accuracy (GA), while DeepSeek scores only 0.16–0.18 on these criteria. This pattern indicates that DeepSeek frequently fails to identify correct writing issues and produces ineffective suggestions.

Both models exhibit high selectivity (WS: 0.91–0.95), indicating they avoid unnecessary criticism. However, for DeepSeek, this occurs alongside extremely low accuracy, giving the impression of a permissive model that avoids criticism by missing issues entirely. This suggests that the model is selectively incorrect rather than effectively discriminative, highlighting that high selectivity does not translate to high-quality feedback when underlying issues remain undetected.

Further differences emerge in suggestion quality. While Qwen maintains moderate clarity and actionability (GC: 0.78), its suggestion specificity (GS) remains relatively low (0.49–0.58), indicating that even when models identify writing short-

comings, they struggle to produce highly targeted guidance. DeepSeek shows a similar pattern, performing poorly across all suggestion dimensions (GS: 0.04, GC: 0.29–0.36), reflecting a lack of focused and actionable feedback.

The choice of prompting approach shows minimal impact for both models. Qwen achieves identical average scores (0.81) with both approaches, while DeepSeek shows marginal variation (0.35–0.38). This suggests that for writing feedback, model capability is a stronger determinant of quality than prompting strategy. Unlike structured domains, writing evaluation appears to be a holistic task that does not benefit substantially from decomposition into specialized subtasks. Table 4 presents evaluation results for the writing domain across 55 research proposals.

Configuration	WA	WS	WC	GA	GS	GC	Avg
Qwen 3.5-U	0.93	0.91	0.82	0.91	0.49	0.78	0.81
Qwen 3.5-MA	0.91	0.93	0.75	0.91	0.58	0.78	0.81
DeepSeek R1-U	0.16	0.95	0.56	0.18	0.04	0.36	0.38
DeepSeek R1-MA	0.18	0.93	0.51	0.18	0.04	0.29	0.35

Table 4: Writing domain results (n=55). U = Unified, MA = Multi-Agent.

3.3 Mathematics Results

DeepSeek R1 with the unified approach achieves the highest overall performance (0.91), outperforming all other configurations. Both models exhibit near-perfect error detection (MA, MS $\approx$ 1.0), indicating that identifying mathematical mistakes is comparatively feasible, as models can follow a step-by-step approach and mathematical steps are more grounded in rules, suggesting that dealing with this level of mathematical mistakes is not a limiting factor for current LLMs.

The main difference lies in the guidance provided in the solution. DeepSeek with unified settings achieves substantially higher solution accuracy (SA: 0.81) than Qwen-Unified (0.25) and both multi-agent configurations (0.03–0.11). This pattern highlights that DeepSeek’s reasoning capabilities are best utilized in unified settings to generate mathematically valid corrective guidance.

It is also observed that, unlike programming and writing, these models are highly sensitive to prompting approaches for mathematics. The multi-agent approach significantly degrades performance for both models, with average scores dropping from 0.79 to 0.75 for Qwen and from 0.91 to 0.70 for

DeepSeek. The degradation is more pronounced in clarity (MC) and solution accuracy (SA), indicating that task decomposition may fragment the coherent reasoning required for effective mathematics feedback. Table 5 presents the evaluation results for the mathematics domain across 150 problems.

Configuration	MA	MS	MC	SA	SS	SC	Avg
Qwen 3.5-U	1.00	1.00	0.52	0.25	1.00	0.99	0.79
Qwen 3.5-MA	1.00	1.00	0.38	0.11	1.00	0.99	0.75
DeepSeek R1-U	0.99	0.99	0.99	0.81	0.89	0.83	0.91
DeepSeek R1-MA	1.00	1.00	0.15	0.03	1.00	0.99	0.70

Table 5: Mathematics domain results (n=150). U = Unified, MA = Multi-Agent.

4 Analysis & Discussion

Results reveal that neither model dominates universally across domains, and that architecture–task alignment is a stronger determinant of feedback quality than parameter count. Qwen 3.5 performs best on writing (avg ≈ 0.81), DeepSeek R1 excels on mathematics in unified mode (avg = 0.91), and programming challenges both models most severely (avg = 0.29–0.48).

Programming. The difficulty of programming feedback reflects a fundamental asymmetry in LLM capabilities: code generation and code understanding are distinct tasks (Haroon et al., 2025). While generation translates intent into code, diagnostic feedback requires reasoning about existing code semantics—a capability that benchmark scores do not capture. Semantic-preserving mutations cause LLMs to fail on previously localized faults in 78% of cases, revealing reliance on syntactic patterns over semantic understanding (Haroon et al., 2025), and systematic failures have been reported when LLMs verify code against specifications (Jin and Chen, 2025). Our findings provide direct educational evidence for this gap: fluency in explaining code does not imply understanding of code behavior. Notably, Qwen 3.5 outperforms DeepSeek on code generation benchmarks, yet DeepSeek achieves 66% higher feedback scores (0.48 vs. 0.29)—confirming that generation capability does not transfer to diagnostic tasks.

Writing. The results reveal a clear performance gap between Qwen 3.5 and DeepSeek (avg 0.81 vs. 0.35–0.38), with the gap most pronounced on accuracy: Qwen correctly identifies writing issues (WA = 0.93) and produces accurate suggestions (GA = 0.91), while DeepSeek scores only 0.16–0.18 on

both. DeepSeek’s high selectivity (WS ≈ 0.94) alongside near-zero accuracy reveals *avoidance behavior*—the model avoids criticism by missing issues entirely rather than correctly filtering them. This shows that high selectivity is not a reliable proxy for feedback quality. Prompting approach has minimal effect on either model, suggesting that writing assessment is a holistic task where model capability matters more than prompting architecture.

Mathematics. Both models achieve near-perfect error detection (MA, MS ≈ 1.0), confirming that identifying mathematical mistakes is tractable at this scale. The key differentiator is solution guidance: DeepSeek-Unified achieves SA = 0.81 vs. only 0.25 for Qwen, reflecting DeepSeek’s reasoning advantage. However, the multi-agent approach severely degrades both models—DeepSeek drops from avg 0.91 to 0.70, with SA collapsing to 0.03. This challenges the assumption that task decomposition improves performance (Wang et al., 2025): mathematics requires a coherent reasoning chain, and splitting error diagnosis from feedback generation disrupts it.

Key takeaway. Multi-agent prompting is domain-conditional: neutral in programming, harmful in mathematics, and inconsequential in writing. It should not be adopted by default for educational feedback generation, and its benefit depends on whether the task is decomposable or requires a cohesive reasoning chain. Beyond prompting, domain characteristics strongly influence feedback quality. Mathematics proves most tractable (avg 0.70–0.91), followed by writing (0.35–0.81), with programming presenting the greatest challenge (0.27–0.48). This hierarchy suggests that structured domains with well-defined, step-by-step correctness criteria are easier to address learners’ issues than tasks that require nuanced judgments. Model architecture is a stronger determinant of feedback quality than parameter count. DeepSeek R1’s reasoning-optimized approach excels in structured domains that require logical analysis, while Qwen 3.5’s general-purpose design works better for holistic evaluation tasks. This pattern cautions against a one-size-fits-all model selection across multiple domains. Finally, our results highlight that programming models can generate fluent feedback while failing to accurately identify bugs, confirming that code-generation capabilities do not transfer to code-diagnostic tasks.

5 Conclusion and Future Directions

We examine a single LLM-based model (as a unified evaluation framework) for feedback generation across three disciplines: (i) mathematics, (ii) academic writing, and (iii) programming. We evaluated two open-source models of different paradigms: reasoning-optimized (DeepSeek R1) and general-purpose (Qwen 3.5) under unified and multi-agent prompting techniques. Our findings reveal that open-source models show potential for cross-domain feedback generation, but with varying effectiveness. DeepSeek R1 excels in structured domains where step-by-step reasoning approaches are favored, such as mathematics (avg 0.91), while Qwen 3.5 performs better on holistic tasks such as writing (avg 0.81). Programming presents a challenge for both models (avg 0.27-0.48). Despite their remarkable code generation capabilities on standard coding benchmarks, they are not yet strong enough for complex code-diagnostic tasks. These results caution against using a single model-supported system across all assessment types, and highlight that model architecture is a stronger determinant of feedback quality than prompting strategy.

Our evaluation assesses feedback quality through human judgment; however, high-quality feedback does not necessarily result in greater learning gains. Future research should examine whether feedback generated by LLMs supports students in developing conceptual understanding and critical thinking skills, rather than directly providing solutions. Following recent work on AI tutoring systems (Murtazin et al., 2025) and students' tendency to rely on AI models for direct answers across writing, programming, and general learning contexts (Denny et al., 2024; Li and Ma, 2025; Ge et al., 2025), we plan to test and deploy our framework as a classroom tool and systematically measure its impact on student performance and learning outcomes.

Limitations

This study has several limitations. First, evaluation is based on offline human judgments rather than classroom deployment, so educational impact on student learning remains unverified, and it is unclear whether high-scoring feedback would help improve learning outcomes, enhance student understanding, and develop critical thinking skills. Second, the evaluation covers only two small

open-source models (Qwen 3.5 27B and DeepSeek R1 32B) and three relatively small datasets: 200 programming samples (ConDefects), 55 writing samples (Exposia), and 150 mathematics samples (MathEDU) datasets, which limits the breadth of generalization. Our primary goal was to demonstrate a constrained environment to maximize the feasibility of replicating the proposed framework. Our findings may not generalize to larger open-source or proprietary models (e.g., GPT, Claude, Gemini). Although the datasets differ in their specific educational contexts and assessment formats, this diversity was intentionally used to assess cross-domain generalizability. Additionally, MathEDU was selected as the closest available dataset to reflect university-level mathematics, due to the absence of suitable calculus or high-level mathematics feedback dataset.

Third, our binary annotation scheme may not capture partial correctness or nuanced differences in feedback quality. Furthermore, our annotators were graduate students with domain expertise but no pedagogical training, which may have influenced their assessments of the pedagogical appropriateness of generated feedback. Inter-annotator agreement was calculated on a limited subset of overlapping samples, which may not fully capture annotation reliability across all domains and the full dataset. Finally, reference supervision differs across domains, especially in programming, where no gold feedback was available.

References

- Areen Alnemrat, Hesham Aldamen, Mohamad Al-mashour, Mutasim Al-Deaibes, and Rami Al-Sharefeen. 2025. [AI vs. teacher feedback on EFL argumentative writing: a quantitative study](#). *Frontiers in Education*, 10.
- Esteban M. Aucejo and Kelvin Wong. 2025. [The effect of feedback on student performance](#). *Journal of Public Economics*, 241:105274.
- Anderson Pinheiro Cavalcanti, Arthur Barbosa, Ruan Carvalho, Fred Freitas, Yi-Shan Tsai, Dragan Gašević, and Rafael Ferreira Mello. 2021. [Automatic feedback in online learning environments: A systematic literature review](#). *Computers and Education: Artificial Intelligence*, 2:100027.
- Jorge Cisneros-González, Natalia Gordo-Herrera, Iván Barcia-Santos, and Javier Sánchez-Soriano. 2025. [JorGPT: Instructor-aided grading of programming assignments with large language models \(LLMs\)](#). *Future Internet*, 17(6):265.

- Paul Denny, Stephen MacNeil, Jaromir Savelka, Leo Porter, and Andrew Luxton-Reilly. 2024. [Desirable characteristics for ai teaching assistants in programming education](#). In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, pages 408–414.
- Bingyu Dong, Jie Bai, Tao Xu, and Yun Zhou. 2024. [Large language models in education: A systematic review](#). In *2024 6th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 131–134. IEEE.
- Darren Paul Fisher, Gaelle Brotto, Iris Lim, and Colette Southam. 2025. [The impact of timely formative feedback on university student motivation](#). *Assessment & Evaluation in Higher Education*, 50(4):622–631.
- Wentao Ge, Yuqing Sun, Ziyang Wang, Haoyue Zheng, Weiyang He, Piaohong Wang, Qianyu Zhu, and Benyou Wang. 2025. [Srlagent: Enhancing self-regulated learning skills through gamification and llm assistance](#). *arXiv preprint arXiv:2506.09968*.
- Jovita Gilmore. 2025. [Going beyond surface-level analysis: Using chatgpt to give feedback and praise for student writing](#). *English Australia Journal*, 41(1):85–90.
- Sghaier Guizani, Tehseen Mazhar, Tariq Shahzad, Wasim Ahmad, Afsha Bibi, and Habib Hamam. 2025. [A systematic literature review to implement large language model in higher education: issues and solutions](#). *Discover Education*, 4(1):35.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shitong Ma, Xiao Bi, and 1 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645:633–638.
- Sabaat Haroon, Ahmad Faraz Khan, Ahmad Humayun, Waris Gill, Abdul Haddi Amjad, Ali R. Butt, Mohammad Taha Khan, and Muhammad Ali Gulzar. 2025. [Assessing the impact of code changes on the fault localizability of large language models](#). *arXiv preprint arXiv:2504.04372*.
- Muhammad Haseeb, Ahmad Imam Amjad, Maaz Amjad, and Victor S. Sheng. 2026. From code to rubrics: A multimodal evaluation of LLM systems for automated feedback generation. In *Companion Proceedings of the ACM on Web Conference (WWW)*. In press.
- Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. [LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13806–13834.
- John Hattie and Helen Timperley. 2007. [The power of feedback](#). *Review of educational research*, 77(1):81–112.
- Wei-Ling Hsu, Yu-Chien Tang, and An-Zi Yen. 2026. [MathEDU: Feedback generation on problem-solving processes for mathematical learning support](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2883–2901, Rabat, Morocco. Association for Computational Linguistics.
- Haolin Jin and Huaming Chen. 2025. [Uncovering systematic failures of LLMs in verifying code against natural language specifications](#). In *Proceedings of the 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, page 3819–3823.
- Annette Kinder, Fiona J. Briese, Marius Jacobs, Niclas Dern, Niels Glodny, Simon Jacobs, and Samuel Leßmann. 2025. [Effects of adaptive feedback generated by a large language model: A case study in teacher education](#). *Computers and Education: Artificial Intelligence*, 8:100349.
- Charles Koutchene, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, Syed Ashraf, and Paul Denny. 2025. [Evaluating language models for generating and judging programming feedback](#). In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education (SIGCSE)*, pages 624–630.
- Yong Oh Lee, Byeonghun Bang, Joohyun Lee, and Sejun Oh. 2026. [Personalized auto-grading and feedback system for constructive geometry tasks using large language models on an online math platform](#). *IEEE Access*, pages 17788–17802.
- Huiyong Li and Boxuan Ma. 2025. [Design of ai-powered tool for self-regulation support in programming education](#). *arXiv preprint arXiv:2504.03068*.
- Michael Pin-Chuan Lin, Jing-Yuan Huang, Daniel H Chang, Gerald Tembrevilla, G Michael Bowen, Eric Poitras, Vasudevan Janarthanan, and Jeeho Ryoo. 2026. [Open-source large language models in education: A narrative review of evidence, pedagogical roles, and learning outcomes](#). In *AI in Education*, page 4. MDPI.
- Naiming Liu, Shashank Sonkar, Zichao Wang, Simon Woodhead, and Richard G. Baraniuk. 2023. [Novice learner and expert tutor: Evaluating math reasoning abilities of large language models with misconceptions](#). *arXiv preprint arXiv:2310.02439*.
- Martin Messer, Neil C. C. Brown, Michael Kolling, and Mia Shi. 2024. [Automated grading and feedback tools for programming education: A systematic review](#). *ACM Transactions on Computing Education (TOCE)*, 24(1):1–43.
- Kristina Murtazin, Oleg Shvets, Karl-Erik Karu, Viljam Puusep, Jelena Vendelin, Martijn Meeter, and Gunnar Piho. 2025. [Student support during project-based learning using simulated automated formative feedback: an experimental evaluation](#). In *Frontiers in Education*, volume 10, page 1621644. Frontiers Media SA.

Fredrick Eneye Tania-Amanda Nkoyo, Chukwue-buka Fortunate Ijezue, Maaz Amjad, Ahmad Imam Amjad, Sabur Butt, and Gerardo Castañeda-Garza. 2025. [Advances in auto-grading with large language models: A cross-disciplinary survey](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 477–498. Association for Computational Linguistics.

Nelly Rigaud Téllez, Patricia Rayón Villela, and Roberto Blanco Bautista. 2024. [Evaluating chatgpt-generated linear algebra formative assessments](#). In *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(5):75–82.

Yaoxiang Wang, Zhiyong Wu, Junfeng Yao, and Jinsong Su. 2025. [Tdag: A multi-agent framework based on dynamic task decomposition and agent generation](#). *Neural Networks*, 185:107200.

Yonghao Wu, Zheng Li, Jie M. Zhang, and Yong Liu. 2023. [Condefects: A new dataset to address the data leakage concern for LLM-based fault localization and program repair](#). *arXiv preprint arXiv:2310.16253*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and Others. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.

Boyang Yang, Haoye Tian, Weiguo Pian, Haoran Yu, Haitao Wang, Jacques Klein, Tegawendé F. Bis-syandé, and Shunfu Jin. 2024. [Cref: An llm-based conversational software repair framework for programming tutors](#). In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*, pages 882–894.

Dennis Zyska, Alla Rozovskaya, Ilia Kuznetsov, and Iryna Gurevych. 2026. [Exposía: Teaching and assessment of academic writing skills for research project proposals and peer feedback](#). *arXiv preprint arXiv:2601.06536*.

Figure A.1. Unified Prompt Template

You are an experienced computer science instructor reviewing a student’s buggy code submission. Your goal is to provide feedback that helps the student understand and fix the bug on their own.

Below is a problem description and an incorrect program written by a student.
 <problem description>
 <language>
 <student code>

Before writing your feedback, reason through the following steps internally:

1. Identify the exact location and type of the primary bug.
2. Determine the conceptual misunderstanding that likely caused it.
3. Decide how to guide the student toward the fix without revealing the solution.

Then write your feedback following these rules:

- Start with one sentence acknowledging something the student did correctly.
- Clearly identify the bug location (e.g., line number) and its error type.
- Explain the underlying conceptual misunderstanding in plain language.
- Describe the fix direction (e.g., “change X to Y”) without providing corrected code.
- Do not flag issues that do not exist in the student’s code.

Output: 150–250 words, 3–4 short paragraphs, no mark-down headers or code blocks.

Figure 2: Unified prompt for programming feedback generation.

Figure A.2. Writing Unified Prompt Template

You are an academic writing expert evaluating a Bachelor's thesis research proposal. Your role combines subject-matter review and writing instruction.

Below is a Bachelor's thesis research proposal written by a student.

<expose text>

Evaluate the proposal across four dimensions using the rubric below. For each criterion:

1. Locate the relevant passage in the text.
2. Apply the rubric to what is explicitly written — do not assume missing content.
3. Assign an integer score within the allowed range.
4. Ensure scores match your written analysis.

Dimensions and maximum scores:

- Problem & RQ (MO1–MO7): max 9 pts
- Literature & Theory (AP1–AP7, BI1–BI2): max 13 pts
- Methodology (AP8, MD1–MD7): max 8 pts
- Writing Quality (LA1–LA2): max 4 pts

Scoring rules: assign only integers within the defined range; score based on development, not mere mention; use only evidence explicitly present in the text.

Output format:

SCORES: MO1:[s] | MO2:[s] | ... | MO7:[s]
AP1:[s] | ... | AP7:[s] | BI1:[s] | BI2:[s]
AP8:[s] | MD1:[s] | ... | MD7:[s] | LA1:[s] | LA2:[s]

FEEDBACK: [150–250 words. Strengths first, then key areas for improvement.]

Figure 3: Unified prompt for academic writing feedback and scoring.

Figure A.3. Mathematics Unified Prompt Template

You are an experienced mathematics tutor reviewing a student's incorrect solution. Your goal is to help the student identify and understand their mistake so they can correct it independently.

Below is a mathematics problem and a student's incorrect solution.

<problem>
<student answer>
<student process>

Before writing your feedback, reason through the following steps internally:

1. Identify the step in the student's process where the error first occurs.
2. Determine whether the error is conceptual, procedural, or computational.
3. Decide how to guide the student without simply giving the answer away.

Then write your feedback following these rules:

- Acknowledge any correct steps the student took before the error, if applicable.
- Clearly identify where and what the mistake is, using the student's own work as reference.
- Explain the mathematical misunderstanding in plain language at the student's level.
- Guide the student step by step and state the correct answer at the end.
- Do not penalize valid alternative solution paths.
- Do not assume reasoning that is not shown in the student's process.

Output: FEEDBACK: [50–120 words. Correct work (if any) → mistake → guidance → correct answer.]

Figure 4: Unified prompt for mathematics feedback generation.

Figure A.4. Multi-Agent Prompt Template (Agent – A)

You are a precise code diagnostician. Your only job is to identify the primary bug in a student's code submission.

Analyze the following student code against the problem statement.

⟨problem description⟩

⟨language⟩

⟨student code⟩

Your tasks are as follows:

1. Identify the single most critical bug that explains why the code fails. If multiple bugs exist, focus on the root cause.
2. Do not provide student-friendly feedback — only a structured technical diagnosis.
3. Output ONLY the following format with no extra commentary:

BUG LOCATION: ⟨line number or segment, e.g. "Line 5: n = n / 2"⟩

ERROR TYPE: ⟨Logic Error | Off-by-One | Incorrect Operator | Wrong Variable | Missing Condition | Type Error | Other⟩

ROOT CAUSE: ⟨one sentence: what the code does vs. what it should do⟩

STUDENT MISCONCEPTION: ⟨one sentence: the likely conceptual gap⟩

FIX DIRECTION: ⟨one sentence: the correction, without writing corrected code⟩

Figure 5: Error Analyzer prompt for the multi-agent programming pipeline.

Figure A.5. Multi-Agent Prompt Template (Agent – B)

You are an experienced computer science instructor. You have received a structured bug analysis for a student's submission. Your task is to convert this technical diagnosis into clear, encouraging, and pedagogically useful feedback.

Below is a bug analysis for a student's code submission.

⟨problem description⟩

⟨language⟩

⟨student code⟩

⟨bug analysis⟩

Your tasks are as follows:

1. Open with one sentence acknowledging something the student did correctly.
2. Pinpoint the bug location and name its error type in plain language.
3. Explain the conceptual misunderstanding behind the bug clearly.
4. Guide the student toward the fix (e.g., "change X to Y") without providing corrected code.

Please follow these rules:

- Ground every claim strictly in the bug analysis — do not invent errors not identified.
- Use accessible language suitable for an introductory programming student.
- Do not write code blocks or corrected code in your output.

Output: 150–250 words, 3–4 short paragraphs, no markdown headers or code blocks.

Figure 6: Feedback Generator prompt for the multi-agent programming pipeline.

Findings of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners

Mariano Felice and Lucy Skidmore

English Language Research & Impact

British Council, UK

name.surname@britishcouncil.org

Abstract

This paper reports findings from the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners across three L1s (Spanish, German and Mandarin). The task featured open and closed tracks, using data from the British Council’s Knowledge-based Vocabulary Lists (KVL). Submissions were received from 23 teams employing diverse modelling approaches, including transformers, Large Language Models, feature-based approaches and ensembles. Results were evaluated using RMSE, with winning systems significantly exceeding the baseline and establishing new state-of-the-art benchmarks. This paper offers an examination of the participating systems, performance across tracks and L1s, and the factors that can affect prediction accuracy.

1 Introduction

Vocabulary is a crucial aspect of language knowledge, shaping what learners can understand and produce. Establishing the difficulty of vocabulary is therefore essential for creating level-appropriate content and developing valid, reliable assessment instruments (Schmitt et al., 2020; François and Gala, 2024). However, determining word difficulty still relies on labour-intensive processes involving expert judgment and costly pretesting, which limits scalability and slows innovation (Dunn et al., 2026). As learning and assessment increasingly rely on digital platforms, the need for more efficient and scalable solutions is more pressing than ever.

While previous shared tasks have explored related problems such as Complex Word Identification (Paetzold and Specia, 2016; Yimam et al., 2018), Lexical Complexity Prediction (Shardlow et al., 2021), Lexical Simplification (Shardlow et al., 2024) and Item Difficulty Estimation (Yaneva et al., 2024), they were not designed with English language learners in mind and did not explore the influence of learners’ L1 on L2 vocabulary diffi-

culty. This shared task addresses that gap by bringing vocabulary difficulty prediction into the domain of language learning and assessment, offering a new perspective on a problem that has traditionally relied on psychometric calibration methods. In doing so, it serves as a testbed for state-of-the-art NLP models, with findings that have the potential to inform AI-powered solutions for item writing, content generation, adaptive testing, and personalised vocabulary learning.

2 Related Research

In the language learning and assessment domain, NLP-based vocabulary difficulty prediction spans a range of approaches, each with its own framing of ‘difficulty’. Such work includes CEFR level classification of vocabulary in isolation (Alfter and Volodina, 2018; Settles et al., 2020) and in context (Aleksandrova and Pouliot, 2023; Bannò et al., 2025), classification of vocabulary difficulty using crowd-sourced ratings as labels (Degraeuwe, 2025) as well as prediction of continuous difficulty values derived from testing learners (Skidmore et al., 2025). This shared task follows the approach of the latter, which sits at the intersection of two areas of research in NLP: Question Difficulty Estimation from Text and Lexical Complexity Prediction.

Question Difficulty Estimation from Text (QDET) aims to predict the difficulty of test items directly from their text, without the need for response data. A growing body of research has explored QDET for high-stakes assessment, driven by its advantages in efficiency and scalability over traditional psychometric approaches (AlKhuzayy et al., 2024). While the latest methods in NLP have been successfully applied in domains related to content knowledge assessment (Yaneva et al., 2024), QDET approaches for language assessment, where difficulty is more closely tied to the linguistic properties of the item, still largely rely on sim-

pler models (AlKhuyaey et al., 2024; Peters et al., 2025). This is particularly true for L2 vocabulary item difficulty estimation, which commonly features ‘vocabulary-only’ datasets that are not well-suited to the latest contextualised word embedding models (Benedetto et al., 2023).

Lexical Complexity Prediction (LCP) is an extension of Complex Word Identification (CWI), which concerns the automatic detection of complex words, primarily to support downstream text simplification (North et al., 2025). LCP expands the binary classification task of CWI to regression, assigning each word a continuous complexity score. The definition of complexity varies by domain, ranging from crowd-sourced ratings to morphosyntactically derived values (North et al., 2023). The majority of LCP and CWI research focuses on L1 vocabulary complexity. However, in line with the increasing body of work advocating for personalised LCP (Lee and Yeung, 2018; Gooding and Tragut, 2022), where demographic features such as L1 background are accounted for, LCP for L2 learners is an increasingly popular focus of research (North and Zampieri, 2023; Degraeuwe, 2025).

3 Shared Task Description

This shared task was organised by researchers at the British Council as part of the BEA Workshop, held at ACL in July 2026. The task addresses vocabulary difficulty prediction for learners of English with diverse first language (L1) backgrounds. Building on prior work (Skidmore et al., 2025), this shared task framed vocabulary difficulty prediction as a regression problem, where systems predict the difficulty of a target English word given a vocabulary test item written in the learner’s L1. Data for the task were drawn from the British Council’s Knowledge-based Vocabulary Lists (KVL) (Schmitt et al., 2021, 2024), a multilingual dataset containing such items with psychometrically calibrated difficulty scores for learners from three L1 backgrounds: Spanish (ES), German (DE) and Mandarin (CN).

Participation was open to the research community following a limited timeline. Training and development data alongside baseline models were released on 26th January 2026, giving participants approximately eight weeks to develop their systems before test data were made available on 20th March. Teams then had one week to submit their final system outputs. Systems were evaluated on the test set by the organisers, with results announced

on 3rd April. Further details on the tracks, data, baseline models and evaluation are outlined below and on the shared task website¹.

3.1 Tracks

The shared task included two tracks:

Closed Track: Designed to elicit models that rely on the provided dataset and standard, publicly available NLP tools. The goal was to explore optimal model design under controlled data conditions.

Data: Systems were permitted to only use the provided training data for each corresponding L1. Additional training data or combining data from different L1s was not allowed.

Models, tools and databases: Publicly available, ‘off-the-shelf’ pre-trained transformer models (e.g. BERT, RoBERTa) and their embeddings, standard NLP tools (e.g. taggers, parsers) and publicly available linguistic databases (e.g. WordNet, frequency lists) were allowed.

LLMs: Publicly available large language models were permitted strictly as a tool to extract features from the item text. For example, generating scalar features (e.g. imageability ratings), textual features (e.g. English translation of the L1 source), or deriving embeddings.

Open Track: The ‘anything goes’ category, which allowed for maximum flexibility and the use of external resources. The purpose of this track was to encourage creativity and explore the full potential of current AI technology.

Data: Any additional training data was permitted. This included public corpora, proprietary datasets, synthetic data, or combining different L1 subsets.

Models, tools and databases: No restrictions.

LLMs: Any system using an LLM to generate a difficulty prediction or identify the target word was classified as open track, regardless of whether the LLM output was used directly or passed to a downstream model.

3.2 Data

The dataset for the task was derived from the British Council’s Knowledge-based Vocabulary Lists (KVL) (Schmitt et al., 2021, 2024), initially developed to collate difficulty rankings of English vocabulary for learners with L1 backgrounds of Spanish, German and Mandarin. The productive

¹britishcouncil.org/data-science-and-insights/bea2026st

English word knowledge of over 100,000 learners was assessed using items testing form-based recall of individual lemmas in a translation format (Laufer and Goldstein, 2004). From approximately 3.3 million responses, difficulty estimates were derived separately for each L1, applying random-item-random-person (RPRI) Rasch models (De Boeck, 2008) built within a generalised linear mixed model (GLMM) framework (Dunn, 2024).

Below is an example test item in Spanish, where learners were required to input the remainder of the target English word “house”²:

```
casa Vivo en una casa grande que tiene tres
dormitorios.
h _ _ _ _
```

For the shared task, participants were provided the ‘Extended KVL Dataset for NLP’ (Skidmore et al., 2025) for model training and development. This dataset contains 6,768 parallel English vocabulary test items, with prompts in the three L1s, totalling 20,304 items. The dataset was split into train and development subsets to facilitate comparison to baseline models for the task, however participants were allowed to use this development data to train their models if desired. For final model evaluation, an additional 748 vocabulary test items per L1 (2,244 total) were released. Table A5 in the Appendix summarises the dataset split. The dataset contained the following columns:

- `item_id`: An ID number from 1 to 6,768. Items with the same `item_id` across different L1s share the same English target word).
- `L1`: The L1 of the prompt (‘es’, ‘de’, or ‘cn’).
- `en_target_word`: The English target word.
- `en_target_pos`: The part of speech of the English target word.
- `en_target_clue`: A partial-spelling clue of the English target word.
- `L1_source_word`: The corresponding L1 source word(s).
- `L1_context`: The L1 contextualising prompt.
- `GLMM_score`: The GLMM difficulty estimate for the vocabulary test item. A lower score indicates a more difficult word.

Details for how to access the shared task data can be found on the British Council’s website.

²The German and Mandarin versions had similar, yet distinct prompts, for example in German: “Haus Ich wohne in einem Haus mit Garten.” And in Mandarin: “房子 我买了一座房子。”

3.3 Baseline Models

The baseline models used for the task are multilingual XLM-RoBERTa-base transformer encoders (Conneau et al., 2020), fine-tuned on the training subset of the data following (Skidmore et al., 2025). The input text includes the question in the same order as it is presented to the test takers (L1 source word, L1 context, EN clue), followed by the target answer (EN target word), as shown in the example:

```
casa </s> Vivo en una casa grande que
tiene tres dormitorios. </s> h____
</s> house
```

Four baseline models were provided to participants, three in the closed track (`baseline_closed_es`, `baseline_closed_de` and `baseline_closed_cn`) which were fine-tuned on the L1 subsets of the data separately, and one in the open track (`baseline_open_xx`) which was fine-tuned on all of the L1 subsets combined. These models are available via GitHub.³ Hyperparameters for the baseline models can be found in Table A6 in the Appendix.

3.4 Evaluation

Submissions were evaluated using two metrics: Root Mean Squared Error (RMSE) and Pearson correlation, chosen for comparability with prior work in LCP and QDET research (North et al., 2023; AlKhuyaey et al., 2024). RMSE was used as the primary metric for the official rankings while Pearson correlation provided a scale-independent measure to facilitate cross-L1 comparison.

4 Participation in the Shared Task

4.1 Teams and Submissions

In total, 23 teams participated in the shared task, from organisations across 16 countries: Bangladesh, Belgium, China, Germany, India, Indonesia, Ireland, Japan, Mexico, Romania, South Korea, Switzerland, United Arab Emirates, United Kingdom, United States of America and Uruguay. Teams were invited to submit up to three ‘runs’ per track and L1. The distribution of teams and runs is summarised in Table 1. Almost all teams submitted to the closed track (22 out of 23), with eight of those teams also submitting to the open track. One team (SurreyCTS) submitted only to the open track. Most teams provided submissions for all three L1s,

³<https://github.com/britishcouncil/bea2026st/>

L1	Open		Closed	
	Teams	Runs	Teams	Runs
Spanish	8	23	21	55
German	8	23	20	52
Mandarin	9	25	21	52
Total	9	71	22	159

Table 1: Number of participating teams and runs.

whereas two teams (AIDA and BZPT) focused on just one L1, Spanish and Mandarin, respectively. Table A7 in the Appendix provides a breakdown of the submitted runs per team.

4.2 Summary of Approaches

Out of the 23 participating teams, 19 published an accompanying system description paper. Table A8 in the Appendix reports the tracks, L1s, approaches, and primary models explored, where submissions are tagged with up to four attributes: **transformers** (transformer-encoder models fine-tuned for prediction), **LLMs** (LLMs fine-tuned for prediction), **features** (any use of features), and **ensemble** (any combination of the above models).

Fine-tuning encoder-only transformer models was the most popular approach, explored by 17 out of 19 teams. The most widely used model family was XLM-RoBERTa (Conneau et al., 2020), particularly XLM-RoBERTa-large, which was used by eight teams, while RemBERT (Chung et al., 2021) and DeBERTA-v3 (He et al., 2023) were also popular choices. Of note, three teams built systems using mmBERT-base, a newer multilingual encoder-only transformer recently shown to outperform the previous generation of multilingual models (Marone et al., 2025). Encoders pretrained for related tasks were also explored, including COMET (Rei et al., 2020) (SurreyCTS) for machine translation quality estimation as well as the sentence embedding models Multilingual E5 (Wang et al., 2024) and BGE-M3 (Chen et al., 2024) (uogal, UOL@IDEM). At the architectural level, some teams moved beyond standard CLS-token pooling, experimenting with methods such as attention-weighted pooling, multi-layer concatenation of hidden states, and target-word-specific pooling (AIDA, Failure, SurreyCTS). Two teams experimented with modified training objectives: Jinnie’s Lab employed multi-task learning, jointly predicting difficulty alongside an auxiliary POS tag; TOEBM employed a multi-objective training framework,

supplementing the standard regression loss with contrastive objectives designed to better align the learned representations with the task structure.

Fine-tuning LLMs was explored by three of the open-track teams (Glite, Sakura, TeamXBC). All three systems used Low-Rank Adaptation (LoRA or QLoRA) (Hu et al., 2022) to fine-tune models from several decoder-only families including Qwen (Yang et al., 2024), Mistral (Jiang et al., 2023; Liu et al., 2026), LLaMA (Dubey et al., 2024), and GLM (Team GLM, 2024). Sakura additionally introduced a novel technique to fine-tune LLMs for continuous value prediction using soft targets.

Feature engineering was also widespread, with 15 of the 19 teams incorporating features into their models, typically fused architecturally with transformer representations (e.g. EduNLP, Failure) or used as inputs to gradient-boosted regressors (e.g. RETUYT-InCo). Some teams applied novel ways to integrate features: AIDA used features to calibrate transformer outputs by predicting their residual error, while uogal augmented transformer inputs with NMT-generated translations of L1 contexts as a complementary cross-lingual signal. Beyond these integration strategies, the features themselves broadly fell into three groups: target word, cross-lingual, and test item. Word-level features included surface properties (e.g. word length), psycholinguistic predictors (e.g. age of acquisition), and L2 learner-specific resources (e.g. CEFR levels) (Boosted Cats, SATLab). Cross-lingual features captured relationships between the L1 source word or context and the English target, most commonly with features related to orthographic and semantic similarity (e.g. RETUYT-InCo). Test item features captured difficulty related to the item content, such as the proportion of letters revealed by the spelling clue (e.g. Data Asgardians, EduNLP) and how recoverable the English target is from the L1 context (UOL@IDEM). Beyond traditional corpus-based and lexical resources, two notable extraction strategies were applied. The first involved probing language models as proxy test-takers, recording signals such as token surprisal and masked token prediction probability (Glite, Sakura). The second used LLMs as direct judges, prompting them to rate aspects of difficulty such as the item’s quality (Glite) or the degree of lexical ambiguity (Sakura).

Model ensembling was adopted by 15 of the 19 teams. The most common strategy was averaging across multiple runs of models from the same

architecture, primarily transformer encoders (e.g. SAAKTH, Token Titans). Exceptions included an array of classical regressors (UGA Threshold) and an ensemble of fine-tuned LLMs (Sakura). Many teams combined architecturally diverse models, typically pairing transformers with feature-based models and weighting predictions toward the transformer (e.g. Data Asgardians, NLP-Explorers). Teams explored stacking ensembles, using a meta-learner such as linear regression or a shallow MLP to combine predictions (e.g. TOEBM, uogal). Three teams (Glite, Sakura, SATLab) used meta-learners that consumed both model predictions and handcrafted item-level features.

Open-track teams took advantage of the relaxed restrictions in several other ways. Three teams fine-tuned decoder-only LLMs for difficulty prediction. Five teams trained models jointly on all three L1s rather than on a separate per-L1 basis. Glite made use of a proprietary sense-level lexical resource with difficulty estimates. One team (uogal) experimented with pretraining on EFCAMDAT (Geertzen et al., 2013), a corpus of L2 English learner essays, though the resulting models were not included in their final system. Moreover, Glite and Sakura used commercial LLM APIs to generate features. Two teams made notable use of agentic AI: TeamXBC used AI coding agents to build their ML solution with varying degrees of human input, while Glite’s autonomous research framework enabled the systematic evaluation of over 1,000 candidate features across 270+ tracked experiments.

5 Results

The official leaderboards for each track and L1 are included in Appendix B, where we report RMSE and Pearson correlation for each run. Given the small differences in RMSE across systems, we performed a statistical significance analysis to identify groups of systems with equivalent performance. Significance testing was conducted using a non-parametric bootstrap method (Efron and Tibshirani, 1993) with 10,000 iterations. To account for multiple comparisons, we applied a Bonferroni correction, setting the significance threshold to $\alpha_{adj} = 0.05/(n - 1)$, where n is the number of submissions within a given track and L1. Systems were grouped sequentially, such that a new group was created only when a system performed significantly worse than the current group’s leader. A summary of the results including

the best entry per team is included in Table 2. Results show that for all tracks and L1s, most teams were able to beat the baseline with at least one of their submissions, often by a large margin.

The winning team for the closed track was Glite (Philippov et al., 2026), whose submitted runs took the top three positions across all three L1s. The best runs achieved an RMSE of 0.903, 0.885 and 0.776 for Spanish, German and Mandarin, respectively, with corresponding Pearson correlations of 0.877, 0.871 and 0.889. The Glite systems used a per-L1 CatBoost regressor (Prokhorenkova et al., 2018), combining scalar out-of-fold predictions from fine-tuned encoder models with hand-crafted features. These were selected from a large candidate pool via Recursive Feature Elimination (RFE) under nested cross-validation, applied separately for each L1. Features included word-level difficulty (e.g., surface form, frequency, morphology, concreteness, CEFR levels), cross-lingual features (e.g., cognates and false friends), and test item difficulty (e.g. spelling difficulty and candidate competition features). Feature extraction relied on traditional corpus-based and lexical resources as well as masked-LM pseudo-likelihood guessing to probe model behaviour as a proxy test-taker. The pool also included a small set of derived second-order features, such as pairwise interactions and model-disagreement metrics. Other systems falling within the top statistical grouping included uogal (German, Mandarin), who used ensembles of fine-tuned transformer models augmented with EN translations of L1 content, and Sakura (Mandarin), who combined a single fine-tuned encoder prediction with explainable features in an XGBoost regressor.

The winning team for the open track was Sakura (Nohejl et al., 2026), whose submitted runs took the top three positions for German and Mandarin, and the top two positions for Spanish. The best runs achieved an RMSE of 0.742, 0.723 and 0.630 for Spanish, German and Mandarin, respectively, with corresponding Pearson correlations of 0.919, 0.916 and 0.928. The approach was centred on a novel soft-target cross-entropy loss for fine-tuning LLMs to predict continuous difficulty values. Three LLMs from different families were fine-tuned using this method and combined via linear regression stacking with out-of-fold predictions. This ensemble was then enhanced with word-level difficulty features (e.g., production and reception frequency, CEFR level, word length), cross-lingual features (e.g., orthographic similarity and morphemic over-

lap), and test item difficulty features (e.g. spelling difficulty and lexical ambiguity, extracted using LLMs as raters). Prompt-based features were calibrated using temperature-scaled, probability-weighted inference (G-Scale) (Nohejl et al., 2025). This configuration was the best-performing for Spanish and Mandarin. The team also experimented with a further set of test item difficulty features combining LLM-prompted difficulty ratings and proxy test-taker estimates, in which LLMs were prompted to solve the items. This configuration took the top position for German. Also within the top statistical grouping, Glite applied the same approach as their closed-track submission, expanding the feature set to include both fine-tuned and prompted LLM difficulty ratings, proxy test-taker estimates, and LLM-based item judgements.

6 Analysis and Discussion

To investigate how performance varied across tracks and L1s, we calculated Mean Absolute Error (MAE), Mean Signed Error (MSE), average prediction Standard Deviation (Avg. SD) and average Pearson correlation (Table 3). Results show that MAE is consistently lower for systems in the open track (by 0.13 on average). This aligns with findings from teams that evaluated joint training across L1s (EduNLP, Jinnie’s Lab, Sakura, TeamXBC), all of which reported improvements over per-L1 training, suggesting the open track gains reflect, at least in part, the benefit of shared signal across L1 subsets. This is most pronounced for Mandarin, which shows the lowest average error despite the orthographical distance from English. However, as noted by Data Asgardians, this is likely due to the reduced range of GLMM values for Mandarin (approximately -4 to 4), which is narrower than for Spanish (approximately -6 to 5) and German (approximately -5 to 4). In fact, looking at average Pearson correlation (a scale-independent measure) reveals that open-track performance for Mandarin is actually lower than for Spanish and German, confirming that its lower MAE is an artefact of the narrower scale.

MSE was positive in all cases, indicating that systems tend to underestimate difficulty on average. Open-track systems show less underestimation across all three L1s, though this is less pronounced for Mandarin — again, likely reflecting its narrower scale. Average SD improves for Spanish and German but increases slightly for Mandarin in the open

track, suggesting that while overall accuracy improves, open-track systems do not achieve greater consensus. Together, these results confirm that the benefits of cross-lingual training are not uniform across L1s.

To examine the L1 and track level differences outlined in Table 3, we investigated individual predictions for each item in the test set. Figure 1 shows the spread of predictions for each test item (in yellow), sorted from easiest (highest GLMM score) to hardest (lowest GLMM score). Ground-truth values are shown in purple, contrasted against regression lines for each winning submission (pink) and mean system performance (dotted red). In addition, we include trend curves for the lowest and highest predictions (dotted orange) in order to better evaluate prediction drift.

System predictions show strong monotonic alignment with true difficulty, meaning that models can successfully capture the ordinal relationship of the target values. This is confirmed by the high Pearson correlation coefficients reported in Appendix B. Predictions overlap substantially with target values across the dataset, except at the extremes, where the items are increasingly difficult to predict. This is regardless of the difficulty range (notably larger for Spanish) and likely a consequence of the small number of samples in the dataset with extreme difficulty values. Several submission papers support this, reporting weaker accuracy at both ends of the GLMM scale. EduNLP found that ‘very hard’ items ($\text{GLMM} < -3$) contribute 25.4% of total squared error despite comprising only 5.1% of items. UOL@IDEM report that the top 20% of ‘very easy’ items are over-predicted by $+1.03$ to $+1.28$ GLMM points by their closed track models. TOEBM showed that prediction bias may be model dependent: XLM-RoBERTa underestimated difficult items, whereas mDeBERTa-v3 showed systematic overestimation and mmBERT was more consistent across difficulty levels. Ensembling with a Ridge regressor exploited these complementary biases to produce more consistent predictions.

Looking at the boundary lines, we observe that the spread of predictions in the closed track is broadly homogenous across all L1s, although Spanish and Mandarin do exhibit an opening towards more difficult items (lower GLMM values), indicating less consensus among systems. Given the small number of extremely difficult items in the datasets, it is unsurprising that systems struggle to predict them. In the open track, prediction spread

Spanish				German				Mandarin			
Rank	Team	RMSE	Pearson	Rank	Team	RMSE	Pearson	Rank	Team	RMSE	Pearson
1	Glite	0.903	.877	1	Glite	0.885	.871	1	Glite	0.776	.889
2	uogal	0.975	.858	2	uogal	0.903	.869	2	Sakura	0.816	.874
3	AIDA	0.976	.857	3	Sakura	0.963	.844	3	uogal	0.820	.879
4	Sakura	0.983	.854	4	NLP-Explorers	0.992	.845	4	TOEBM	0.880	.853
5	NLP-Explorers	1.040	.839	5	SAAKTH	0.994	.844	5	NLP-Explorers	0.882	.861
6	SAAKTH	1.045	.836	6	TOEBM	0.997	.832	6	UOL@IDEM	0.891	.860
7	TOEBM	1.063	.826	7	Jinnie's Lab	1.011	.827	7	SAAKTH	0.900	.860
8	Jinnie's Lab	1.084	.818	8	SATLab	1.036	.830	8	Jinnie's Lab	0.905	.843
9	RETUYT-InCo	1.094	.843	9	UOL@IDEM	1.037	.834	9	SATLab	0.921	.846
10	SATLab	1.108	.813	10	UZHCL	1.071	.826	10	TeamXBC	0.968	.851
11	UZHCL	1.121	.816	11	TeamXBC	1.114	.837	11	UZHCL	0.991	.839
12	UOL@IDEM	1.132	.813	12	Data Asgardians	1.117	.830	12	Data Asgardians	1.006	.854
13	TeamXBC	1.146	.823	13	Boosted Cats	1.118	.783	13	Boosted Cats	1.030	.790
14	Token Titans	1.170	.812	14	Failure	1.118	.788	14	UGA Threshold	1.031	.791
15	EduNLP	1.176	.788	15	EduNLP	1.124	.793	15	Mixed Signals	1.055	.790
16	Data Asgardians	1.182	.820	16	UGA Threshold	1.166	.761	16	EduNLP	1.058	.800
17	Failure	1.219	.763	17	Mixed Signals	1.173	.784	17	Failure	1.090	.763
18	Boosted Cats	1.236	.755	18	BASELINE	1.258	.773	18	RETUYT-InCo	1.106	.754
19	UGA Threshold	1.236	.758	19	RETUYT-InCo	1.260	.713	19	BASELINE	1.140	.753
20	Mixed Signals	1.244	.760	20	Unibuc	1.305	.692	20	Unibuc	1.176	.717
21	BASELINE	1.257	.765	21	Token Titans	1.564	.760	21	Token Titans	1.382	.718
22	Unibuc	1.421	.664					22	BZPT	1.588	.590

(a) Closed Track.

Spanish				German				Mandarin			
Rank	Team	RMSE	Pearson	Rank	Team	RMSE	Pearson	Rank	Team	RMSE	Pearson
1	Sakura	0.742	.919	1	Sakura	0.723	.916	1	Sakura	0.630	.928
2	Glite	0.754	.916	2	Glite	0.764	.905	2	Glite	0.660	.920
3	TeamXBC	0.876	.885	3	TeamXBC	0.826	.888	3	TeamXBC	0.722	.904
4	uogal	0.975	.858	4	uogal	0.903	.869	4	uogal	0.820	.879
5	SurreyCTS	1.034	.839	5	SurreyCTS	0.945	.854	5	SurreyCTS	0.861	.863
6	Jinnie's Lab	1.053	.829	6	Jinnie's Lab	0.990	.834	6	Jinnie's Lab	0.885	.851
7	SATLab	1.087	.819	7	SATLab	1.010	.828	7	SATLab	0.926	.847
8	EduNLP	1.143	.802	8	EduNLP	1.071	.816	8	EduNLP	0.992	.824
9	BASELINE	1.198	.783	9	BASELINE	1.166	.786	9	BASELINE	1.034	.804
								10	BZPT	1.123	.744

(b) Open Track.

Table 2: Best results per team, track and L1.

Track	L1	MAE	MSE	Avg. SD	Pearson
Closed	ES	.897	.108	.627	.773
Closed	DE	.869	.214	.607	.795
Closed	CN	.770	.214	.523	.815
Open	ES	.744	.061	.439	.850
Open	DE	.722	.098	.448	.849
Open	CN	.685	.155	.529	.826

Table 3: Aggregated metrics per track and L1.

shrinks for all L1s, especially for Spanish and German. For Mandarin, however, dispersion increases at both ends of the difficulty scale in the open track, suggesting that additional signals may introduce noise rather than useful information for this L1 at the extremes. This finding may in part be due to the greater orthographical distance between Mandarin and English limiting cross-lingual generalisability compared to Spanish and German, reported in prior work (Skidmore et al., 2025) and corroborated in submissions for the shared task (UGA Threshold, Boosted Cats, SurreyCTS).

The wide vertical dispersion observed for some items indicates cases where the systems largely disagree. These are words that the systems find particularly confusing. Likewise, items with the highest error (where the yellow predictions are furthest away from the purple values) indicate the words that were the most difficult to predict. Table 4 includes some examples. The words that are easier to predict are either cognates, concrete nouns, or specific terminology, whereas the most difficult ones tend to be polysemous words (whose most frequent sense is not the target sense), abstract words, collocates or words with specific parts-of-speech (gerunds, adverbs, verbs). Item-level error analyses carried out by the teams corroborate this, with models consistently struggling most with words with uncommon senses or grammatical roles. In particular, EduNLP found that nominalised ‘-ing’ forms showed a 58% higher RMSE than other words, such as “dining” “baking”, “speaking” seen in Table 4, while Boosted Cats found that items where the tested POS differed from the dominant POS,

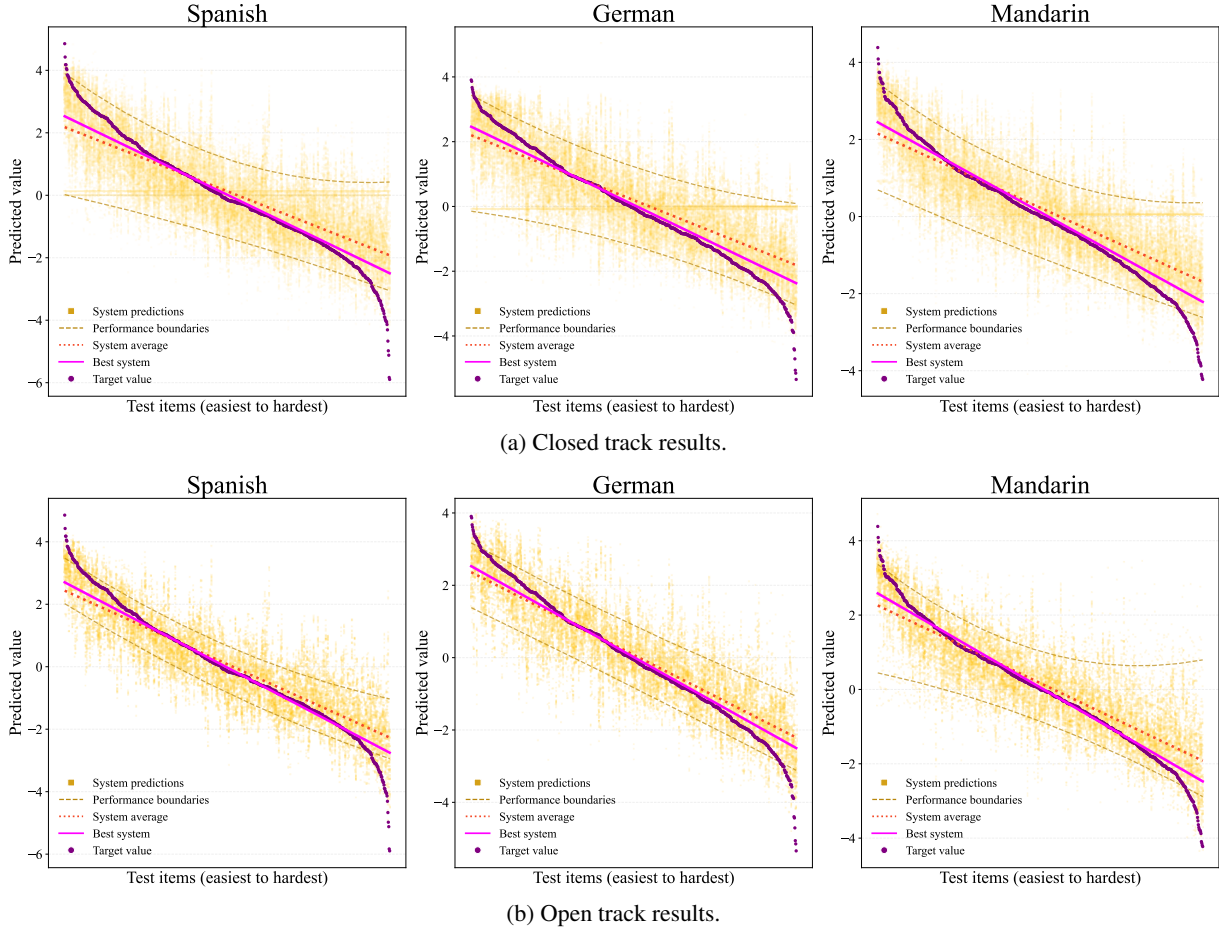


Figure 1: System prediction distributions sorted by target difficulty (higher GLMM scores $\rightarrow$ easier items).

illustrated by “spring”, “amount” and “received” in Table 4, increased prediction variance by 24–32%.

Considering the shared task approaches as a whole, there are several conclusions that can be drawn. Firstly, improvements over the XLM-RoBERTa-base baseline were driven primarily by model choice. In particular, switching to XLM-RoBERTa-large delivered improvements across multiple teams (e.g. TeamXBC). Architectural modifications such as multi-layer pooling and layer aggregation offered smaller improvements in some systems (Jinnie’s Lab, SurreyCTS), though more complex modifications such as layerwise learning rate decay were largely ineffective or harmful (TeamXBC). Secondly, feature-based models were competitive with the baseline, particularly for Spanish and German where cross-linguistic similarity signals are informative (Boosted Cats, SAAKTH). However, ablation studies for hybrid models revealed that incorporating hand-crafted features provided only small improvements to already strong ensemble baselines (Glite, SAAKTH). Nevertheless, feature-based models offer a clear advantage

in interpretability, providing insights into what drives difficulty for different learner groups, which is essential for item development in practice (Dunn et al., 2026). Finally, ensembling almost always helped, with model diversity proving more important than raw individual model strength (TOEBM, uogal). Interestingly, however, complex stacking strategies were found to overfit to the small development set provided (EduNLP, TeamXBC, uogal).

The two winning teams, Glite and Sakura, shared a common architectural approach to the task: combining out-of-fold predictions from multiple fine-tuned models with a broad set of hand-crafted features. Both were among the few to use features targeting item-level as well as word-level properties, in particular proxy test-taker signals from masked LMs and LLM-based judgements of item difficulty. They were among only three teams to experiment with fine-tuned and prompted LLMs as a core component of their systems, which proved highly effective but remains relatively underexplored across the shared task as a whole. Both systems also used cross-validation strategies for model training. To-

Category	Spanish	German	Mandarin
Least Confusing (lowest std dev)	breathing (n.), prepared (adj.), product (n.), demand (v.), defeat (v.)	notice (n.), necessary (adj.), pile (n.), reward (v.), admire (v.)	necessary (adj.), morality (n.), blame (v.), criminal (adj.), surrounding (adj.)
Most Confusing (highest std dev)	shape (v.), marked (adj.), dozen (num.), received (adj.), knight (v.)	crowned (v.), tweet (v.), received (adj.), walker (n.), shape (v.)	standing (n.), schoolmate (n.), remarkably (adv.), clear (adv.), house (v.)
Easiest to predict (lowest error)	perfectionist (n.), bow (n.), frequently (adv.), blame (v.), crowd (v.)	necessary (adj.), increasingly (adv.), dog (n.), genetically (adv.), album (n.)	surgery (n.), left (adj.), exercise (v.), silence (n.), luxury (n.)
Hardest to predict (highest error)	dining (n.), baking (n.), received (adj.), clear (adv.), unchallenged (adj.)	streak (n.), anger (v.), umbrella (n.), fund (n.), hydrated (adj.)	speaking (n.), stake (n.), swear (v.), clear (adv.), might (v.)

Table 4: Sample vocabulary items for different prediction categories.

gether, these factors suggest that the performance gap between the winning systems and the rest was driven by the combination of multiple model predictions, broad and well-chosen feature coverage, and evidence-based model development.

Across submissions, a picture emerged of vocabulary difficulty as shaped by both an underlying component tied to features of the English word and a L1-specific signal. This is supported by the fact that language-agnostic pipelines with no L1-specific features outperformed the baseline models (Failure). Improvements from joint L1 training (EduNLP, Jinnie’s Lab, Sakura, TeamXBC) further confirm that much of this can be shared across learner groups, which supports findings of previous research (Skidmore et al., 2025). At the same time, per-L1 hand-crafted features also proved valuable, and analyses of feature importance revealed meaningful differences in what drives difficulty for different learner groups (Boosted Cats, Sakura). A similar pattern was evident in ensemble models, where different encoder combinations were preferred for different L1s (uogal). Cross-lingual transfer experiments reinforced this picture: Spanish and German models transferred to each other reasonably well, but Mandarin transferred poorly, reflecting fundamental differences in how difficulty is structured across typologically distant pairs (Boosted Cats, SAAKTH). Together, these findings affirm the value of L1-aware modelling with clear implications for scalable, personalised assessment and adaptive vocabulary learning.

7 Future Work

Looking ahead, several directions emerge from the findings of the shared task. The ‘compression effect’ of the target labels provide motivation to experiment with alternative labelling strategies, such as ordinal or banded difficulty rather than continuous GLMM scores. The L1 asymmetry between European languages and Mandarin makes a strong case for extending vocabulary difficulty prediction

to more typologically diverse L1s, such as Arabic or Japanese, to test whether the findings here generalise beyond the specific language pairs studied. Finally, making use of learner response data to validate the insights produced by interpretable models would provide empirical grounding for conclusions that currently rest on model behaviour alone. Features identified as important by multiple systems such as spelling difficulty, polysemy of the target word, and cross-lingual transfer signals could be directly tested against learner responses, representing a valuable bridge between the NLP and language testing research communities.

8 Conclusions

This paper presented an overview of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners, predicting difficulty scores for English lexical items across three L1s (Spanish, German, and Mandarin). The task comprised closed and open tracks, restricting and permitting additional data and models respectively. A total of 23 teams participated using a variety of approaches, including classical machine learning, transformer models, LLMs and ensembles. Most systems outperformed our transformer-based baseline, with winners in each track doing so by a large margin.

System analysis showed that ensembles combining linguistic features with fine-tuned model predictions achieved the best results. Systems tended to underestimate difficulty and struggled the most with polysemous words. Open-track systems consistently outperformed their closed-track counterparts, suggesting that joint training across L1s improves predictive accuracy. However, the degree of impact of these approaches varies across L1s and difficulty levels and warrants further investigation.

We believe the insights gained from the shared task will significantly contribute to the design of more robust vocabulary prediction models, ultimately supporting better test item development and more adaptive language learning applications.

9 Limitations

Most of the limitations of this work stem from the datasets used to train and evaluate the systems. The data was restricted to three L1s (German, Spanish, Mandarin), so conclusions about system performance must be interpreted with caution, as they may not generalise to learners with different linguistic backgrounds. Furthermore, the test items across these L1s were not standardised, potentially introducing variation in difficulty estimates beyond learner background. Some participants found that the small development set (677 items per L1) limited reliable model selection and ensemble calibration, which may have impacted the overall performance of some systems. In addition, the use of the probabilistic values derived from the GLMM framework as observed difficulty values comes with its own caveats, as discussed by [Schmitt et al. \(2024\)](#).

Since GLMM scores are estimations rather than direct observations, RMSE values should be interpreted with care. To provide a more robust assessment of model performance, we also report Pearson correlation, which captures the linear relationship between predictions and true values and is less sensitive to numerical fluctuations.

The GLMM difficulty values in our datasets are also not directly comparable across L1s, since they were derived from different population samples. This is something that must be taken into account when training systems using a combination of data from all L1s, such as our open track baseline.

For limitations pertaining to individual submissions, please refer to the corresponding description papers.

10 Ethics statement

The dataset for this shared task was collected by the British Council through a large-scale crowdsourcing initiative ([Schmitt et al., 2021](#)). Participation was voluntary and participants agreed for their response data to be used for research purposes. All data was anonymised prior to analysis. Vocabulary difficulty scores were derived from aggregated responses using psychometric modelling, ensuring that individual identities cannot be inferred.

A primary objective of this shared task is the promotion of responsible AI practices in language education, with an emphasis on transparency, reproducibility, and fairness. For this reason, all the data, baseline models, evaluation scripts and documenta-

tion have been made publicly available for research and educational purposes. The insights from the submitted models and their respective description papers also aim to foster greater transparency and explainability in the domain of AI for education.

Acknowledgements

The authors would like to thank the original research team behind the Knowledge-based Vocabulary Lists, whose datasets have been fundamental to this shared task. Special thanks go to Karen J. Dunn for her continued guidance and support. We also wish to thank all participants of the BEA 2026 Shared Task for their engagement and contributions.

References

- Abid Al Hossain and Kamruzzaman Khan Alve. 2026. Failure at BEA 2026 Shared Task 1: One pipeline, three L1s: A unified language-agnostic system for vocabulary difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Desislava Aleksandrova and Vincent Pouliot. 2023. [CEFR-based contextual lexical complexity classifier in English and French](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 518–527, Toronto, Canada. Association for Computational Linguistics.
- David Alfter and Elena Volodina. 2018. [Towards single word lexical complexity prediction](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 79–88, New Orleans, Louisiana. Association for Computational Linguistics.
- Samah AlKhuzayy, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2024. [Text-based question difficulty prediction: A systematic review of automatic approaches](#). *International Journal of Artificial Intelligence in Education*, 34(3):862–914.
- Stefano Bannò, Kate M. Knill, and Mark J. F. Gales. 2025. [Exploiting the English vocabulary profile for L2 word-level vocabulary assessment with LLMs](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 632–646, Vienna, Austria. Association for Computational Linguistics.
- 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 Comput. Surv.*, 55(9).

- Yves Bestgen. 2026. SATLab at BEA 2026 Shared Task 1: Predicting the difficulty of English words for three L1 learners using primarily psycholinguistic features. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Xiaobin Chen. 2026. TeamXBC at BEA 2026 Shared Task 1: How AI (and I) won the shared task: Vibe and agentic coding solutions for practical machine learning problems. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Seok Hyeon Cho, JunHyeok Choi, Sangeun Ji, and Sung Won Han. 2026. AIDA at BEA 2026 Shared Task 1: A two-stage framework for L1-aware vocabulary difficulty prediction with representation diversity and residual calibration. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Hyung Won Chung, Thibault F  vry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. [Rethinking embedding coupling in pre-trained language models](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzm  n, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Emma Dalbo. 2026. UGA Threshold at BEA 2026 Shared Task 1: Predicting vocabulary acquisition difficulty with hand-crafted SLA-based features. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Paul De Boeck. 2008. [Random item IRT models](#). *Psychometrika*, 73(4):533–559.
- Jasper Degraeuwe. 2025. [You shall know a word’s difficulty by the family it keeps: Word family features in personalised word difficulty classifiers for L2 Spanish](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 312–325, Vienna, Austria. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and 1 others. 2024. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Karen J. Dunn. 2024. [Random-item Rasch models and explanatory extensions: A worked example using L2 vocabulary test item responses](#). *Research Methods in Applied Linguistics*, 3(3):100143.
- Karen J. Dunn, Lucy Skidmore, and Thomas Rogers. 2026. [When measurement meets machine learning: interpretability and scalability in modelling item difficulty for language assessment](#). *Frontiers in Education*, 11.
- Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.
- Thomas Fran  ois and N  ria Gala. 2024. [Graded resources for learning and teaching foreign languages: An overview](#). *ITL – International Journal of Applied Linguistics*, 175(1):8–24.
- 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 (SLRF)*, pages 240–254.
- Sian Gooding and Manuel Tragut. 2022. [One size does not fit all: The case for personalised word complexity models](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 353–365, Seattle, United States. Association for Computational Linguistics.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.
- 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*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, and 1 others. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.
- Nouran Khallaf and Serge Sharoff. 2026. UOL@IDEM at BEA 2026 Shared Task 1: Neural fusion and feature-rich modeling for L1-aware vocabulary difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.

- Tayyab Latif, Asifa Bibi, Sabur Butt, and Sidorov Grigori. 2026. NLP-Explorers at BEA 2026 Shared Task 1: DeBERTa–CatBoost weighted ensemble approach for L1-specific vocabulary difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Batia Laufer and Zahava Goldstein. 2004. [Testing vocabulary knowledge: Size, strength, and computer adaptiveness](#). *Language learning*, 54(3):399–436.
- John Lee and Chak Yan Yeung. 2018. [Personalizing lexical simplification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Wicaksono Leksono, Joanito Agili Lopo, Tsamara R. N. Nugraha, Ahmad Cahyono Adi, and Muhamad Oriza Nurfajri. 2026. TOEBM at BEA 2026 Shared Task 1: Improving lexical difficulty prediction with context-aligned contrastive learning and ridge ensembling. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Zhe Li, Pauline Aguinale, and Jinnie Shin. 2026. Jinnie’s Lab at BEA 2026 Shared Task 1: Precalibration of vocabulary item difficulty with multilingual transformers and multi-task learning. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. [Ministral 3](#). *Preprint*, arXiv:2601.08584.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. [mmBERT: A modern multilingual encoder with annealed language learning](#). *Preprint*, arXiv:2509.06888.
- Jonas Mayer Martins, Zhuojing Huang, Aaricia Herygers, and Lisa Beinborn. 2026. BoostedCats at BEA 2026 Shared Task 1: What makes a word hard to learn? Modeling L1 influence on English vocabulary difficulty. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Karthik Mattu, Adit Dhall, Arshad Naguru, Shubh Sehgal, Thejas Nagesh Gowda, and Hakyung Sung. 2026. SAAKTH at BEA 2026 Shared Task 1: L1-aware English vocabulary difficulty prediction with hybrid transformer and psycholinguistic features. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2025. [A Japanese dataset and efficient multilingual LLM-based methods for Lexical Simplification and Lexical Complexity Prediction](#). *Journal of Natural Language Processing*, 32(4):1129–1188.
- Adam Nohejl, Xuanxin Wu, Yusuke Ide, Maria Angelica Riera Machin, and Yi-Ning Chang. 2026. Sakura at BEA 2026 Shared Task 1: What makes vocabulary difficult? In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2025. [Deep learning approaches to lexical simplification: A survey](#). *Journal of Intelligent Information Systems*, 63:111–134.
- Kai North and Marcos Zampieri. 2023. [Features of lexical complexity: insights from L1 and L2 speakers](#). *Frontiers in Artificial Intelligence*, 6.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. [Lexical complexity prediction: An overview](#). *ACM Comput. Surv.*, 55(9).
- 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.
- Anubhab Parashar and Sandeep Mathias. 2026. Token Titans at BEA 2026 Shared Task 1: Multilingual lexical complexity prediction via fine-tuned XLM-RoBERTa with ensemble decoding. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Sydney Peters, Nan Zhang, Hong Jiao, Ming Li, and Tianyi Zhou. 2025. [Review of text-based approaches to item difficulty modeling in large-scale assessments](#). In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Coordinated Session Papers*, pages 37–47, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).
- Vassili Philippov, Dmitrii Andreev, Pavel Katunin, and Anton Nikolaev. 2026. Glite at BEA 2026 Shared Task 1: Holistic difficulty models dominate, feature engineering closes the gap in L1-aware vocabulary

- difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Adrian Pineda, Sabur Butt, and Héctor Gibrán Ceballos Cancino. 2026. Data Asgardians at BEA 2026 Shared Task 1: A hybrid transformer–feature ensemble for L1-aware English vocabulary difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, pages 6639–6649, Red Hook, NY, USA. Curran Associates Inc.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Santiago Robaina, Aiala Rosá, and Luis Chiruzzo. 2026. RETUYT-INCO at BEA 2026 Shared Task 1: Feature-enriched mDeBERTa for word difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. [Introducing Knowledge-based Vocabulary Lists \(KVL\)](#). *Tesol Journal*, 12(4).
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. [Knowledge-based Vocabulary Lists](#). University of Toronto Press, Toronto.
- Norbert Schmitt, Paul Nation, and Benjamin Kremmel. 2020. [Moving the field of vocabulary assessment forward: The need for more rigorous test development and validation](#). *Language Teaching*, 53(1):109–120.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine learning–driven language assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Avinash Kumar Sharma. 2026. EduNLP at BEA 2026 Shared Task 1: Multi-model ensemble with feature-augmented transformers for vocabulary difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Bernardo Stearns, John P. McCrae, Thomas Gaillat, and Jefkine Kafunah. 2026. uogal at BEA 2026 Shared Task 1: Ensemble of multilingual encoders with NMT augmentation for L1-aware vocabulary difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Team GLM. 2024. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. [Multilingual e5 text embeddings: A technical report](#). *Preprint*, arXiv:2402.05672.
- Georgina Jennifer Willoughby, Jordan Painter, Diptesh Kanojia, Emily Frances Wells, and Constantin Orasan. 2026. SurreyCTS at BEA 2026 Shared Task 1: Semantic funnelling and entropy-based multilingual lexical difficulty prediction. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Victoria Yaneva, Kai North, Peter Baldwin, Le An Ha, 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), pages 470–482, Mexico City, Mexico. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, and 1 others. 2024. Qwen2.5: A party of foundation models. *arXiv preprint arXiv:2412.15115*.

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 Supplementary tables

L1	Train	Development	Test
Spanish	6,091	677	748
German	6,091	677	748
Mandarin	6,091	677	748

Table A5: An overview of the dataset splits used in the shared task, totalling 7,516 vocabulary items per L1 background.

Model name	L1	Learning rate	Weight decay	Warmup ratio	Batch size	Dropout rate
baseline_closed_es	ES	3e-5	0.1	0.1	32	0.1
baseline_closed_de	DE	3e-5	0.0	0.1	32	0.1
baseline_closed_cn	CN	3e-5	0.1	0.1	32	0.1
baseline_open_xx	XX	3e-5	0.1	0.1	32	0.1

Table A6: Optuna hyperparameters for the baseline models, following [Skidmore et al. \(2025\)](#).

Team	Open			Closed		
	ES	DE	CN	ES	DE	CN
AIDA	–	–	–	3	–	–
BZPT	–	–	2	–	–	1
Boosted Cats (HuDS lab)	–	–	–	3	3	3
Data Asgardians	–	–	–	3	3	3
EduNLP	3	3	3	3	3	3
Failure	–	–	–	1	1	1
Glite	3	3	3	3	3	3
Jinnie’s Lab	3	3	3	3	3	3
Mixed Signals	–	–	–	3	3	3
NLP-Explorers	–	–	–	3	3	3
RETUYT-InCo	–	–	–	2	1	1
SAAKTH	–	–	–	3	3	3
SATLab	3	3	3	3	3	3
Sakura	3	3	3	3	3	3
SurreyCTS	3	3	3	–	–	–
TOEBM	–	–	–	3	3	3
TeamXBC	3	3	3	3	3	3
Token Titans	–	–	–	2	3	2
UGA Threshold	–	–	–	3	3	3
UOL@IDEM	–	–	–	3	3	3
UZHCL	–	–	–	1	1	1
Unibuc	–	–	–	2	2	2
uogal	2	2	2	2	2	2
Total	23	23	25	55	52	52

Table A7: Number of runs submitted per team, track, and L1.

Team Name	Track	L1	Approach(es)	Model(s)
AIDA (Cho et al., 2026)	Closed	Spanish	Transformers; features; ensemble	mDeBERTa-v3-base; RemBERT; XLM-RoBERTa-large; LightGBM
Boosted Cats (HuDS lab) (Martins et al., 2026)	Closed	All	Features	CatBoost
BZPT	Both	Mandarin	—	—
Data Asgardians (Pineda et al., 2026)	Closed	All	Transformers; features	XLM-RoBERTa-large
EduNLP (Sharma, 2026)	Both	All	Transformers; features; ensemble	XLM-RoBERTa-base; LightGBM
Failure (Al Hossain and Alve, 2026)	Closed	All	Transformers; features; ensemble	XLM-RoBERTa-base
Glite (Philippov et al., 2026)	Both	All	Transformers; features; ensemble; LLMs	CatBoost; XLM-RoBERTa-base; XLM-RoBERTa-large; XLM-RoBERTa-XL; XLM-RoBERTa-XXL; RemBERT; ELECTRA-base; mBERT; LLaMA-3.1-8B; Qwen2.5-7B
Jinnie’s Lab (Li et al., 2026)	Both	All	Transformers	mmBERT-base
Mixed Signals	Closed	All	—	—
NLP-Explorers (Latif et al., 2026)	Closed	All	Transformers; features; ensemble	DeBERTa-v3-large; XLM-RoBERTa-large; CatBoost
RETUYT-InCo (Robaina et al., 2026)	Closed	All	Transformers; features; ensemble	XGBoost; mDeBERTa-v3-base
SAAKTH (Mattu et al., 2026)	Closed	All	Transformers; features; ensemble	XLM-RoBERTa-large; XGBoost
Sakura (Nohejl et al., 2026)	Both	All	Transformers; features; ensemble; LLMs	mmBERT-base; GLM-4-32B; Qwen2.5-32B; Ministral-3-14B; XGBoost; linear regression
SATLab (Bestgen, 2026)	Both	All	Transformers; features; ensemble	LightGBM; XLM-RoBERTa-base
SurreyCTS (Willoughby et al., 2026)	Open	All	Transformers; features; ensemble	RemBERT; COMET
TeamXBC (Chen, 2026)	Both	All	Transformers; ensemble; LLMs	XLM-RoBERTa-large; RemBERT; Qwen2.5-7B
TOEBM (Leksono et al., 2026)	Closed	All	Transformers; ensemble	XLM-RoBERTa-base; mDeBERTa-v3-base; mmBERT-base
Token Titans (Parashar and Mathias, 2026)	Closed	All	Transformers; ensemble	XLM-RoBERTa-large
UGA Threshold (Dalbo, 2026)	Closed	All	Features; ensemble	scikit-learn models: Ridge; SGD (ElasticNet); Random Forest; Gradient Boosting; Extra Trees; MLP
Unibuc	Closed	All	—	—
uogal (Stearns et al., 2026)	Both	All	Transformers; features; ensemble	Multilingual-E5-large; XLM-RoBERTa-large; RemBERT; InfoXLM-Large
UOL@IDEM (Khallaf and Sharoff, 2026)	Closed	All	Transformers; features	Multilingual-E5-large; BGE-M3
UZHCL	Closed	All	Transformers; ensemble	XLM-RoBERTa-large

Table A8: Participating teams and chosen approaches. The models reported include those used to make predictions (including ensemble members), but exclude models used for feature extraction. Any additional experiments conducted by the teams beyond the submitted runs are not included.

B Official leaderboards

Group	Rank	Team Name	Prediction File	RMSE	Pearson
1	1	Glite	predictions_run_3.csv	0.903	0.877
	2	Glite	predictions_run_2.csv	0.910	0.876
	3	Glite	predictions_run_1.csv	0.920	0.872
2	4	uogal	8enc_opt_weights_..._dev_preds.csv	0.975	0.858
	5	AIDA	predictions_run_1.csv	0.976	0.857
	6	uogal	8enc_dmeta_elasticnet_..._preds.csv	0.977	0.859
	7	Sakura	predictions_closed_max.csv	0.983	0.854
3	8	AIDA	predictions_run_2.csv	1.039	0.840
	9	NLP-Explorers	predictions_run_2.csv	1.040	0.839
	10	NLP-Explorers	predictions_run_1.csv	1.040	0.839
	11	NLP-Explorers	predictions_run_3.csv	1.041	0.838
	12	SAAKTH	predictions_run_3.csv	1.045	0.836
	13	SAAKTH	predictions_run_2.csv	1.053	0.835
	14	TOEBM	predictions_run_3.csv	1.063	0.826
	15	AIDA	predictions_run_3.csv	1.066	0.835
	16	Jinnie's Lab	mbert-base_mtl_mlp_cls_es_preds.csv	1.084	0.818
	17	SAAKTH	predictions_run_1.csv	1.087	0.819
	18	TOEBM	predictions_run_2.csv	1.089	0.816
	19	RETUYT-InCo	predictions_ensemble.csv	1.094	0.843
	20	SATLab	Dev.csv	1.108	0.813
	21	SATLab	Boot.csv	1.110	0.811
	22	Jinnie's Lab	mbert-base_mtl_mlp_mean_es_preds.csv	1.114	0.807
4	23	UZHCL	grand_ensemble_ft_preds.csv	1.121	0.816
	24	UOL@IDEM	run_1.csv	1.132	0.813
	25	UOL@IDEM	run_2.csv	1.134	0.808
	26	UOL@IDEM	run_3.csv	1.140	0.813
	27	Jinnie's Lab	mbert-base_mtl_mlp_mha_1_es_preds.csv	1.142	0.796
	28	TeamXBC	TeamXBC_predictions_run_2.csv	1.146	0.823
	29	Sakura	predictions_explainable.csv	1.156	0.789
	30	Token Titans	predictions_bert_1.csv	1.170	0.812
	31	EduNLP	predictions_closed_ensemble.csv	1.176	0.788
	32	EduNLP	predictions_closed_stretched.csv	1.178	0.788
	33	Data Asgardians	predictions_hp_solo.csv	1.182	0.820
5	34	Data Asgardians	predictions_hp_hybrid.csv	1.186	0.825
	35	Data Asgardians	predictions_closed_solo.csv	1.190	0.814
	36	TeamXBC	TeamXBC_predictions_run_1.csv	1.192	0.816
	37	SATLab	Nonly.csv	1.212	0.778
	38	Failure	predictions_run_1.csv	1.219	0.763
	39	EduNLP	predictions_closed_transformer.csv	1.220	0.787
	40	Boosted Cats	submission_es_run1.csv	1.236	0.755
	41	Boosted Cats	submission_es_run3.csv	1.236	0.755
	42	UGA Threshold	predictions_run_1.csv	1.236	0.758
	43	Boosted Cats	submission_es_run2.csv	1.237	0.754
	44	Mixed Signals	num_es_base_ipa_pos_preds.csv	1.244	0.760
	45	BASELINE	baseline_closed_track.csv	1.257	0.765
	46	UGA Threshold	predictions_run_3.csv	1.261	0.746
	47	Mixed Signals	numeric_es_base_preds.csv	1.264	0.752
	48	UGA Threshold	predictions_run_2.csv	1.268	0.740
	49	Mixed Signals	num_es_base_ipa_pos_mean_preds.csv	1.278	0.744
	50	Sakura	predictions_traditional.csv	1.305	0.721
6	51	RETUYT-InCo	predictions_xgboost.csv	1.323	0.713
	52	TOEBM	predictions_aya_llamma_ens.csv	1.324	0.719
	53	Unibuc	predictions_run_2.csv	1.421	0.664
	54	Unibuc	predictions_run_1.csv	1.425	0.659
7	55	TeamXBC	TeamXBC_predictions_run_3.csv	1.882	0.093
	56	Token Titans	predictions_debert_2.csv	1.885	0.043

Table A9: Full results for Closed Track: Spanish.

Group	Rank	Team Name	Prediction File	RMSE	Pearson
1	1	Glite	predictions_run_3.csv	0.885	0.871
	2	Glite	predictions_run_1.csv	0.887	0.871
	3	Glite	predictions_run_2.csv	0.895	0.868
	4	uogal	8enc_dmeta_elasticnet_..._preds.csv	0.903	0.869
	5	uogal	8enc_opt_weights_..._dev_preds.csv	0.944	0.865
2	6	Sakura	predictions_closed_max.csv	0.963	0.844
	7	NLP-Explorers	predictions_run_3.csv	0.992	0.845
	8	NLP-Explorers	predictions_run_1.csv	0.992	0.845
	9	NLP-Explorers	predictions_run_2.csv	0.993	0.845
	10	SAAKTH	predictions_run_3.csv	0.994	0.844
	11	TOEBM	predictions_run_3.csv	0.997	0.832
	12	SAAKTH	predictions_run_1.csv	1.000	0.831
	13	Jinnie's Lab	mbert-base_mtl_mlp_Mean_..._preds.csv	1.011	0.827
	14	SAAKTH	predictions_run_2.csv	1.012	0.843
3	15	Jinnie's Lab	mbert-base_mtl_mlp_Max_..._preds.csv	1.032	0.818
	16	SATLab	Boot.csv	1.036	0.830
	17	UOL@IDEM	run_2_freq.csv	1.037	0.834
	18	Jinnie's Lab	mbert-base_mtl_mlp_cls_de_preds.csv	1.039	0.815
	19	SATLab	Dev.csv	1.045	0.826
	20	UZHCL	grand_ensemble_ft_preds.csv	1.071	0.826
	21	TOEBM	predictions_run_2.csv	1.076	0.811
	22	UOL@IDEM	run_3_surprisal.csv	1.078	0.818
	23	UOL@IDEM	run_1_allfeatures.csv	1.079	0.819
	24	SATLab	Nonly.csv	1.103	0.812
	25	TeamXBC	TeamXBC_predictions_run_1.csv	1.114	0.837
	26	Data Asgardians	predictions_hp_hybrid.csv	1.117	0.830
	27	Boosted Cats	submission_de_run2.csv	1.118	0.783
4	28	Failure	predictions_run_1.csv	1.118	0.788
	29	Boosted Cats	submission_de_run1.csv	1.118	0.782
	30	TeamXBC	TeamXBC_predictions_run_2.csv	1.121	0.835
	31	Boosted Cats	submission_de_run3.csv	1.121	0.781
	32	EduNLP	predictions_closed_ensemble.csv	1.124	0.793
	33	EduNLP	predictions_closed_stretched.csv	1.125	0.793
	34	Sakura	predictions_explainable.csv	1.126	0.779
	35	Data Asgardians	predictions_closed_hybrid.csv	1.140	0.829
	36	UGA Threshold	predictions_run_3.csv	1.166	0.761
	37	Mixed Signals	numeric_ipa_de_1frz_preds.csv	1.173	0.784
	38	Mixed Signals	num_de_base_ipa_pos_meanings_preds.csv	1.174	0.784
	39	UGA Threshold	predictions_run_1.csv	1.176	0.757
	40	Data Asgardians	predictions_hp_solo.csv	1.177	0.822
5	41	Mixed Signals	numeric_de_base_ipa_pos_preds.csv	1.180	0.773
	42	UGA Threshold	predictions_run_2.csv	1.181	0.753
	43	Sakura	predictions_traditional.csv	1.195	0.747
	44	EduNLP	predictions_closed_transformer.csv	1.202	0.789
	45	TOEBM	predictions_aya_llamma_ensemble.csv	1.237	0.735
6	46	BASELINE	baseline_closed_track.csv	1.258	0.773
	47	RETUYT-InCo	predictions_xgboost.csv	1.260	0.713
	48	Unibuc	predictions_run_1.csv	1.305	0.692
	49	Unibuc	predictions_run_2.csv	1.309	0.692
7	50	Token Titans	predictions_run_1.csv	1.564	0.760
	51	Token Titans	predictions_ensemble.csv	1.569	0.772
	52	Token Titans	predictions_run_2.csv	1.569	0.772
8	53	TeamXBC	TeamXBC_predictions_run_3.csv	1.796	0.299

Table A10: Full results for Closed Track: German.

Group	Rank	Team Name	Prediction File	RMSE	Pearson
1	1	Glite	predictions_run_2.csv	0.776	0.889
	2	Glite	predictions_run_3.csv	0.785	0.886
	3	Glite	predictions_run_1.csv	0.788	0.885
	4	Sakura	predictions_closed_max.csv	0.816	0.874
	5	uogal	8enc_dmeta_elasticnet_..._preds.csv	0.820	0.879
2	6	uogal	8enc_opt_weights_..._dev_preds.csv	0.841	0.880
	7	TOEBM	predictions_run_3.csv	0.880	0.853
	8	NLP-Explorers	predictions_run_2.csv	0.882	0.861
	9	NLP-Explorers	predictions_run_1.csv	0.883	0.860
	10	NLP-Explorers	predictions_run_3.csv	0.885	0.859
	11	UOL@IDEM	run_2_freq.csv	0.891	0.860
3	12	SAAKTH	predictions_run_3.csv	0.900	0.860
	13	Jinnie's Lab	mbert-base_mtl_mlp_Max_..._cn_preds.csv	0.905	0.843
	14	SAAKTH	predictions_run_2.csv	0.911	0.859
	15	SAAKTH	predictions_run_1.csv	0.913	0.851
	16	UOL@IDEM	run_3_top1.csv	0.919	0.858
	17	Sakura	predictions_explainable.csv	0.920	0.837
	18	SATLab	Dev.csv	0.921	0.846
	19	SATLab	Boot.csv	0.924	0.843
	20	UOL@IDEM	run_1_allfeatures.csv	0.930	0.856
	21	TOEBM	predictions_run_2.csv	0.930	0.834
	22	Jinnie's Lab	mbert-base_mtl_mlp_mean_cn_preds.csv	0.932	0.832
4	23	TeamXBC	TeamXBC_predictions_run_1.csv	0.968	0.851
	24	UZHCL	grand_ensemble_ft_preds.csv	0.991	0.839
	25	TeamXBC	TeamXBC_predictions_run_2.csv	1.002	0.859
	26	Data Asgardians	predictions_hp_hybrid.csv	1.006	0.854
	27	TeamXBC	TeamXBC_predictions_run_3.csv	1.006	0.839
	28	Data Asgardians	predictions_hp_solo.csv	1.008	0.850
	29	Data Asgardians	predictions_closed_solo.csv	1.013	0.842
	30	Boosted Cats	submission_cn_run3.csv	1.030	0.790
	31	UGA Threshold	predictions_run_1.csv	1.031	0.791
	32	Boosted Cats	submission_cn_run1.csv	1.034	0.789
	33	TOEBM	predictions_aya_llamma_ensemble.csv	1.037	0.789
	34	Boosted Cats	submission_cn_run2.csv	1.039	0.786
5	35	Mixed Signals	num_ipa_cn_lfrz_5b_preds.csv	1.055	0.790
	36	EduNLP	predictions_closed_ensemble.csv	1.058	0.800
	37	EduNLP	predictions_closed_stretched.csv	1.059	0.800
	38	UGA Threshold	predictions_run_3.csv	1.061	0.778
	39	Mixed Signals	numeric_cn_base_ipa_pos_preds.csv	1.062	0.792
	40	Mixed Signals	num_cn_base_ipa_pos_mean_preds.csv	1.062	0.777
	41	UGA Threshold	predictions_run_2.csv	1.072	0.770
	42	Sakura	predictions_traditional.csv	1.078	0.767
	43	Failure	predictions_run_1.csv	1.090	0.763
	44	SATLab	Nonly.csv	1.094	0.826
	45	RETUYT-InCo	predictions_xgboost.csv	1.106	0.754
	46	EduNLP	predictions_closed_transformer.csv	1.112	0.801
6	47	BASELINE	baseline_closed_track.csv	1.140	0.753
	48	Unibuc	predictions_run_1.csv	1.176	0.717
	49	Unibuc	predictions_run_2.csv	1.178	0.715
	50	Jinnie's Lab	mbert-base_mtl_mlp_mha_1_cn_preds.csv	1.185	0.710
7	51	Token Titans	predictions_run2.csv	1.382	0.718
8	52	Token Titans	predictions_run1.csv	1.436	0.777
9	53	BZPT	predictions_bert.csv	1.588	0.590

Table A11: Full results for Closed Track: Mandarin.

Group	Rank	Team Name	Prediction File	RMSE	Pearson
1	1	Sakura	predictions_finetuned_llms_plus.csv	0.742	0.919
	2	Sakura	predictions_open_max.csv	0.743	0.920
	3	Glite	predictions_run_2.csv	0.754	0.916
	4	Glite	predictions_run_1.csv	0.755	0.916
	5	Sakura	predictions_finetuned_llms.csv	0.760	0.915
2	6	Glite	predictions_run_3.csv	0.837	0.896
	7	TeamXBC	TeamXBC_predictions_run_3.csv	0.876	0.885
3	8	uogal	8enc_opt_weights_..._dev_preds.csv	0.975	0.858
	9	uogal	8enc_dmeta_elasticnet_..._preds.csv	0.977	0.859
	10	TeamXBC	TeamXBC_predictions_run_2.csv	1.003	0.846
4	11	SurreyCTS	predictions_run_3.csv	1.034	0.839
	12	SurreyCTS	predictions_run_1.csv	1.046	0.833
	13	Jinnie's Lab	mbert-base_mtl_mlp_mha_1_xx_preds.csv	1.053	0.829
	14	Jinnie's Lab	mbert-base_mtl_mlp_mean_xx_preds.csv	1.056	0.829
	15	Jinnie's Lab	mbert-base_mtl_mlp_Mean_..._xx_preds.csv	1.061	0.826
5	16	SurreyCTS	predictions_run_2.csv	1.083	0.829
	17	SATLab	Dev.csv	1.087	0.819
	18	SATLab	Mean.csv	1.088	0.819
	19	SATLab	Boot.csv	1.091	0.817
	20	TeamXBC	TeamXBC_predictions_run_1.csv	1.114	0.832
	21	EduNLP	predictions_open_ensemble.csv	1.143	0.802
	22	EduNLP	predictions_open_improved_es.csv	1.147	0.801
	23	EduNLP	predictions_open_mega_stretched.csv	1.150	0.804
6	24	BASLINE	baseline_open_track.csv	1.198	0.783

Table A12: Full results for Open Track: Spanish.

Group	Rank	Team Name	Prediction File	RMSE	Pearson
1	1	Sakura	predictions_open_max.csv	0.723	0.916
	2	Sakura	predictions_finetuned_llms_plus.csv	0.726	0.915
	3	Sakura	predictions_finetuned_llms.csv	0.731	0.914
2	4	Glite	predictions_run_2.csv	0.764	0.905
	5	Glite	predictions_run_1.csv	0.769	0.904
	6	Glite	predictions_run_3.csv	0.790	0.898
3	7	TeamXBC	TeamXBC_predictions_run_3.csv	0.826	0.888
4	8	uogal	8enc_dmeta_elasticnet_..._preds.csv	0.903	0.869
5	9	uogal	8enc_opt_weights_..._dev_preds.csv	0.944	0.865
	10	SurreyCTS	predictions_run_3.csv	0.945	0.854
	11	TeamXBC	TeamXBC_predictions_run_2.csv	0.947	0.856
	12	SurreyCTS	predictions_run_1.csv	0.968	0.844
	13	Jinnie's Lab	mbert-base_mtl_mlp_mean_xx_preds.csv	0.990	0.834
	14	Jinnie's Lab	mbert-base_mtl_mlp_mha_1_xx_preds.csv	1.002	0.830
6	15	SurreyCTS	predictions_run_2.csv	1.004	0.844
	16	Jinnie's Lab	mbert-base_mtl_mlp_Mean_..._xx_preds.csv	1.009	0.827
	17	SATLab	Boot.csv	1.010	0.828
	18	SATLab	Mean.csv	1.014	0.827
	19	SATLab	Dev.csv	1.022	0.824
	20	TeamXBC	TeamXBC_predictions_run_1.csv	1.042	0.848
7	21	EduNLP	predictions_open_ensemble.csv	1.071	0.816
8	22	EduNLP	predictions_open_mega_stretched.csv	1.089	0.815
9	23	BASLINE	baseline_open_track.csv	1.166	0.786
10	24	EduNLP	predictions_open_improved_de.csv	1.397	0.671

Table A13: Full results for Open Track: German.

Group	Rank	Team Name	Prediction File	RMSE	Pearson
1	1	Sakura	predictions_finetuned_llms_plus.csv	0.630	0.928
	2	Sakura	predictions_open_max.csv	0.631	0.927
	3	Sakura	predictions_finetuned_llms.csv	0.640	0.925
	4	Glite	predictions_run_1.csv	0.660	0.920
	5	Glite	predictions_run_2.csv	0.662	0.920
2	6	Glite	predictions_run_3.csv	0.679	0.916
3	7	TeamXBC	TeamXBC_predictions_run_3.csv	0.722	0.904
4	8	uogal	8enc_dmeta_elasticnet_..._preds.csv	0.820	0.879
	9	TeamXBC	TeamXBC_predictions_run_2.csv	0.830	0.880
5	10	uogal	8enc_opt_weights_..._dev_preds.csv	0.841	0.880
	11	SurreyCTS	predictions_run_3.csv	0.861	0.863
	12	SurreyCTS	predictions_run_1.csv	0.883	0.852
6	13	SurreyCTS	predictions_run_2.csv	0.885	0.866
	14	Jinnie's Lab	mbert-base_mtl_mlp_mha_1_xx_preds.csv	0.885	0.851
	15	Jinnie's Lab	mbert-base_mtl_mlp_mean_xx_preds.csv	0.895	0.847
	16	Jinnie's Lab	mbert-base_mtl_mlp_Mean_..._xx_preds.csv	0.899	0.845
	17	SATLab	Boot.csv	0.926	0.847
	18	SATLab	Mean.csv	0.926	0.847
	19	SATLab	Dev.csv	0.933	0.845
7	20	TeamXBC	TeamXBC_predictions_run_1.csv	0.943	0.864
	21	EduNLP	predictions_open_ensemble.csv	0.992	0.824
	22	EduNLP	predictions_open_mega_stretched.csv	1.007	0.825
8	23	BASELINE	baseline_open_track.csv	1.034	0.804
	24	BZPT	predictions_tool-augmented.csv	1.123	0.744
9	25	EduNLP	predictions_open_improved_cn.csv	1.421	0.623
10	26	BZPT	predictions_prompting.csv	1.845	0.044

Table A14: Full results for Open Track: Mandarin.

SATLab at BEA 2026 Shared Task 1: Predicting the Difficulty of English Words for Three L1 Learners Using Primarily Psycholinguistic Features

Yves Bestgen

Statistical Analysis of Text Laboratory (SATLab)

Faculté de psychologie, de logopédie, de sexologie et des sciences de la famille

Université catholique de Louvain

Place Cardinal Mercier, 10 1348 Louvain-la-Neuve, Belgium

yves.bestgen@uclouvain.be

Abstract

This paper presents SATLab’s participation in the BEA 2026 shared task on predicting the difficulty of English words for L2 learners. The proposed system uses features mainly derived from word frequency lists, lexical norms, and psychometric data, which are input into a gradient boosting decision tree model. It outperformed the Baseline system but performed significantly worse than the top-performing teams. Feature contributions to model performance are analysed using gain scores and Spearman rank correlations, and a brief analysis of the most significant errors is provided.

1 Introduction

This paper presents SATLab’s participation in the British Council’s Shared Task at BEA 2026, *Vocabulary Difficulty Prediction for English Learners*, whose aim is to foster the development of regression models for predicting the difficulty of English words for learners of three first languages: Spanish, German, and Mandarin. Such models could, for example, be used to tailor content for learners at different proficiency levels.

To build high-performing models for this task, as in many other areas of NLP, the use of pre-computed embeddings and LLMs has proven most effective (Skidmore et al., 2025; Yaneva et al., 2024). However, it is worth considering whether other features, such as word frequency lists, lexical norms, and psychometric data, can rival the simplest of these approaches and contribute to greater effectiveness. This seems particularly justified because Shardlow et al. (2021) observed, in a shared task on assessing word difficulty, little difference in effectiveness between deep learning approaches and feature-based approaches such as the system developed by Mosquera (2021). Previously, Culligan (2015) found that frequencies from very large corpora provide the best estimate of word difficulty (see also Settles et al. (2020)).

For several years, SATLab has specialised in using these indices to solve complex tasks such as predicting eye saccades during reading or identifying offensive content and hate speech in languages with few linguistic resources (Bestgen, 2021a,b,c). The vocabulary difficulty prediction task is particularly relevant to this objective due to the sustained interest it has attracted in linguistics (Schmitt and Schmitt, 2020) and the availability of ‘fine-grained’ resources based on extensive linguistic knowledge. The system proposed by SATLab is based on these features, which are fed into a gradient boosting decision tree model. One version of the system uses only these psycholinguistic features (*Nonly* version). The other version adds them to the difficulty score predicted by the organisers’ Baseline system, which uses pre-computed embeddings (Felice and Skidmore, 2026; Skidmore et al., 2025).

The next section presents the main characteristics of the shared task. The features, the most important part of the proposal, are then described in detail. Finally, the results of the challenge are reported, along with an analysis to determine which features are the most useful.

2 Data and Task

The data for this challenge is taken from the *Extended Knowledge-based Vocabulary Lists Dataset for NLP* (Schmitt et al., 2021; Skidmore et al., 2025). It consists of 7,516 words, for which the difficulty level for English learners was estimated using a translation task involving over 100,000 learners from three L1 backgrounds. For each target lemma, the test item included the first letter, the length and the part of speech of the target word in English, one or more translations of the word into the L1, and a sentence in the L1 featuring the word in the relevant context. Based on responses in each L1, a difficulty level was estimated using a psychometric procedure (see Schmitt et al. (2024)).

for further details).

The material was divided into three sets: the Learning set ($N = 6,091$), the Development set ($N = 677$), and the Test set ($N = 748$). The task was offered in two tracks. For the closed track, LLMs were not permitted to train the system or combine several L1s. All methods were permitted for the open track. The performance metric is the Root Mean Square Error (RMSE). Each team was allowed to submit three runs for each track and each L1.

The organisers also provided, as the baseline, the system they presented at BEA 2025 (Skidmore et al., 2025), which uses a fine-tuned transformer based architecture. This system was used to predict the Development and Test sets from the Learning set of the shared task materials.

3 Feature Sets

3.1 Features Derived Only from the Items

- Simple statistics: the character length of the target word (TargetL), the source word in L1 (SourceL) and the context (ContextL), the POS tag of the target (PosTag), the first letter of the target (FirstLetter) and the number of proposed L1 words (NbrWordL1).
- Sim12: The orthographic (Levenshtein distance) and phonetic (Soundex algorithm¹) similarity were calculated to estimate the degree of cross-linguistic similarity between the English target word and the L1 source word (Otwinowska and Szweczyk, 2019). These measures are calculated only for German and Spanish.

3.2 Features Extracted from Generic Resources

- FBfr: Herdagdelen and Marelli (2017) Facebook frequency lists for American English and British English based on approximately one billion tokens for each variety. These lists were extracted from publicly available English posts collected between November 2014 and January 2015.
- TWfr: Herdagdelen and Marelli (2017) Rovereto Twitter Corpus frequency on more than one billion tokens from 75 million tweets, collected between December 2010 and July 2011.

- HALFr: Lund and Burgess (1996) Hyper-space Analogue to Language (HAL) frequency norms for approximately 40,000 words provided by (Balota et al., 2007).
- GoogleFr: The Google Ngram frequency word list².
- USENETfr: Shaoul and Chris (2006) USENET Orthographic Frequencies derived from a corpus of more than seven billion words from posts collected between October 2005 and August 2006.
- BNCfr: The frequency of orthographic forms in the 100-million-word British National Corpus. Two frequencies were calculated: the total frequency of the orthographic form and the frequency taking into account the PosTag.
- BasicV: Graham et al. (1993) list of 850 words most frequently used by children in their writing³. These words are classified into 5 grades according to their degree of difficulty for children, with the most difficult words to spell highlighted.
- Glasgow: Scott et al. (2019) lexical norms for age of acquisition, arousal, concreteness, dominance, familiarity, valence, gender association, imageability and semantic size. These norms include 5,553 English words evaluated on average by 33 native English speakers.
- ELP: Balota et al. (2007) lexical characteristics and behavioural measures from lexical decision and naming tasks for more than 40,000 words as available in the English Lexicon Project.
- Latent: Knoph et al. (2024) latent lexical dimensions of Frequency, Complexity, Proximity, Polysemy, and Diversity for 2,060 high-frequency words from the General Service List, 1,051 general academic words from the Academic Word List, and 3,413 domain-specific words from the Academic Vocabulary List.

It may seem surprising to use so many frequency-related features that are more or less strongly correlated. However, this approach has proved particularly effective in the Cognitive Modeling and Computational Linguistics (CMCL) Shared Task on Eye-Tracking Data Prediction (Bestgen, 2021a), enabling the system to outperform all deep learning-based systems participating in the challenge.

¹<https://www.archives.gov/research/census/soundex>

²<https://github.com/hackerb9/gwordlist>

³<https://www.readingrockets.org/topics/writing/articles/basic-spelling-vocabulary-list>

3.3 Features Extracted from L2 Language Learner Resources

- Oxford: The Oxford University Press list of the 3,000 most essential words for English language learners⁴, classified according to their level of difficulty based on CEFR levels A1 to B2.
- L2fr: The frequency of orthographic forms in three corpora of learners of English as a foreign language: the Ten-thousand English Compositions of Chinese Learners Corpus (TECCL), the International Corpus Network of Asian Learners of English (ICNALE) and the First Certificate in English examination scripts (FCE).
- LexCom: Maddela and Xu (2018) lexicon of 15,000 frequent English words annotated for complexity by eleven fluent English speakers of different native languages. These words are drawn from the most frequent words in the Google Ngram Corpus.
- BKL2: Brysbaert et al. (2020) database identifying the English words most familiar to non-native speakers, compiled through a large-scale crowdsourcing study involving 62,000 words.

4 System

LightGBM free software (Ke et al., 2017), a well-known implementation of the gradient boosting decision tree approach, was used to build the regression model. The various parameters were optimised using Optuna. When the feature derived from the Baseline was not used (*Nonly* version), nine-fold cross-validation was used on the development and training sets to select the best model and determine the number of iterations (early stopping rule). Nine folds were used to maintain the ratio between the number of instances in the development and training sets.

When this feature was used, the criterion was performance on the Development set. In the *Dev* version of the system, the Development set was used as provided by the organisers, i.e. the 677 unique instances. In the *Boot* version, 100 bootstrap samples (with replacement) of 677 instances from this Development set were constructed at the start of the optimisation, and the performance of a set of parameters corresponded to the average

performance across these 100 samples. As these two versions produced almost identical results during the test phase, the remainder of this paper will no longer refer to the *Boot* version, which is less conventional.

5 Analysis and Results

5.1 Official Results

The official challenge results are presented in the organisers' summary paper (Felice and Skidmore, 2026). In both tracks and across the three L1s, the *Dev* version performed significantly better (mean RMSE = 1.019) than the Baseline (mean RMSE = 1.176), but it also performed significantly worse than the top-performing team (mean RMSE = 0.790). In the closed track, the *Nonly* version, which does not use pre-computed embeddings, was also more effective (mean RMSE = 1.136) than the Baseline (mean RMSE = 1.218), which relies on a fine-tuned transformer-based architecture. However, the *Nonly* version was considerably less effective than the *Dev* version, which uses the Baseline as an additional feature (mean RMSE = 1.025).

5.2 Feature Contribution to Model Performance

To assess the usefulness of the features for predicting lexical complexity, Table 1 presents the importance scores of the top ten features in each of the three models analysed and for the three languages. These are gain scores, indicating how much each feature contributes to model performance. The table shows the sum of the gains for all features derived from the same resource. To facilitate interpretation, the values correspond to the proportion of each score relative to the maximum score for that model. A value of 0.50 therefore indicates that this feature set is half as useful as the most useful feature set.

For the *Nonly* version, the most useful predictors are clearly Sim12, BKL2, and Oxford. The effectiveness of Sim12 likely stems from its ability to estimate cross-linguistic similarity between the English target word and the L1 source word (Otwinowska and Szewczyk, 2019). The inclusion of Oxford in this group is notable because, as mentioned above, it is a list of just 3,000 words divided into four categories. Although relatively old and not specific to L2 learners, ELP is also very useful. Regarding the frequency lists, those from Twitter and Facebook are by far the best. I must admit

⁴<https://www.oxfordlearnersdictionaries.com/wordlists/oxford3000-5000>

Spanish		German		Mandarin	
Feature	Gain	Feature	Gain	Feature	Gain
Closed <i>Nonly</i>					
Sim12	1.00	Sim12	1.00	BKL2	1.00
BKL2	0.60	BKL2	0.66	Oxford	0.80
Oxford	0.41	Latent	0.44	TWfr	0.40
ELP	0.27	ELP	0.26	ELP	0.38
Glasgow	0.24	Glasgow	0.17	FBfr	0.30
LexCom	0.16	FBfr	0.10	Latent	0.24
Latent	0.13	TWfr	0.10	Glasgow	0.18
TWfr	0.11	LexCom	0.09	LexCom	0.18
FBfr	0.07	TargetL	0.04	SourceL	0.15
BNCfr	0.05	BNCfr	0.04	BNCfr	0.10
Closed <i>Dev</i>					
BL	1.00	BL	1.00	BKL2	1.00
Sim12	0.72	Sim12	0.68	BL	0.85
BKL2	0.42	BKL2	0.33	ELP	0.52
Glasgow	0.15	ELP	0.15	TWfr	0.39
ELP	0.12	Oxford	0.14	FBfr	0.28
Oxford	0.12	TWfr	0.10	Oxford	0.25
TWfr	0.12	Glasgow	0.09	Latent	0.25
LexCom	0.08	FBfr	0.08	BNCfr	0.24
FBfr	0.07	Latent	0.06	Glasgow	0.18
Latent	0.07	LexCom	0.05	LCFreq	0.17
Open <i>Dev</i>					
BL	1.00	BL	1.00	BL	1.00
Sim12	0.58	Sim12	0.41	BKL2	0.54
BKL2	0.39	BKL2	0.26	TWfr	0.25
Glasgow	0.13	Glasgow	0.06	ELP	0.25
Oxford	0.10	ELP	0.05	Latent	0.21
TWfr	0.09	TWfr	0.04	LCFreq	0.17
ELP	0.08	Latent	0.04	Glasgow	0.17
LexCom	0.06	LexCom	0.03	LexCom	0.15
FBfr	0.06	FBfr	0.02	FBfr	0.15
Latent	0.04	LCFreq	0.01	BNCfr	0.10

Table 1: Importance scores (LightGBM gain)

that I do not know why the length of the L1 source (SourceL) is relatively important for Mandarin. At first glance, I thought it would be the length of the target text, as is the case with German.

When the Baseline is introduced into the model (*Dev* versions), it is almost always the best predictor. This is particularly true for the Open track, where it performs much better than the other predictors. In general, there is little difference between the two tracks, which is to be expected since the scores to be predicted are the same and the predictors are the same, except for the Baseline, which was not trained on the same data. Finally, the differences observed between the three L1s do not appear to be very significant.

5.3 Spearman rank correlation with the lexical complexity score

The importance scores presented in Table 1 are not independent. An index that is relatively effective at predicting lexical complexity may be masked by

the presence of other, more effective and strongly correlated indices, such as frequency norms derived from different corpora.

To assess the significance of each feature set independently, Table 2 presents the Spearman rank correlation (ρ) between the features and the score to be predicted, for the ten most highly correlated feature sets. Only the best feature of each set is taken into account. Rank correlation is used because it is insensitive to monotonic transformations, as is the gradient boosting decision tree approach, and is therefore not affected by the choice of one monotonic transformation over another (e.g. logarithmic, square root, inverse, or even highly implausible ones).

In each of the three L1s, the correlations of the best predictors are very similar. However, these correlations are significantly higher in Mandarin than in the other two languages. This result may be linked the fact that most teams performed significantly better in the official competition results for Mandarin than for Spanish and German. The excellent quality of Oxford is particularly evident in Mandarin, where it is the most correlated feature with vocabulary complexity.

LexCom also warrants special attention. Table 1 shows that this predictor does not make a significant contribution to model performance. Correlation analysis, however, reveals that for all three L1s, it is the second-best predictor and very close to the best. Another notable observation is that Sim12 is clearly not among the most highly correlated predictors for Spanish, yet it is the most useful feature for the model in the *Nonly* version and the second most important in the *Dev* versions. This is likely because it provides information distinct from the other features, almost all of which are linked in some way to frequency in the language. Regarding frequency indices, we observe that frequencies derived from Google Ngrams are never among the top ten predictors, nor are those from Usenet and the BNC. Further analysis would be valuable to understand the reasons for these poor performances.

5.4 Analysis of the most significant errors

The previous section identifies the most useful features for each L1. However, this should not obscure the fact that the proposed system makes relatively significant errors. The aim of this section is to gain insights into potential patterns found in poorly estimated items. To this end, I have extracted, sep-

Spanish		German		Mandarin	
Feature	ρ	Feature	ρ	Feature	ρ
Glasgow	0.48	BKL2	0.49	Oxford	0.63
LexCom	0.48	LexCom	0.48	LexCom	0.61
BKL2	0.47	Glasgow	0.47	BKL2	0.61
Oxford	0.43	Sim12	0.46	TWfr	0.60
TWfr	0.42	TWfr	0.44	Latent	0.60
FBfr	0.41	Oxford	0.43	FBfr	0.59
Sim12	0.40	FBfr	0.43	L2fr	0.58
Latent	0.37	ELP	0.39	Glasgow	0.56
ELP	0.37	Latent	0.38	ELP	0.52
BasicV	0.32	BasicV	0.32	HALfr	0.50

Table 2: Spearman rank correlation (Rho)

arately for each L1, the 20 items with the highest absolute error values.

Firstly, there is very little overlap between the three L1s, as over 90% of the items are specific to a single L1. The most significant errors are almost equal to 5, whereas the scale of the judgements to be predicted ranges roughly from -6 to +5.

In 90% of these 60 items, the error is an underestimation of the item’s difficulty. A detailed analysis of the most problematic items shows that the issue often arises because the target is a relatively rare usage of a word well-known in another grammatical category. Examples include ‘clear’ used as an adverb, which usually appears in idiomatic expressions; ‘received’ used as an adjective meaning ‘generally accepted’; ‘very’ used as an emphatic adjective (‘the very nature of’); and ‘dinning’ used as a noun.

Among the most serious errors is also an item in which the word from the source language does not appear verbatim in the Spanish sentence provided in context: *6842,lamp,noun,l____,faro,Se encendió un foco de forma automática.,-2.171975368*

6 Conclusion

The aim of this study was to propose features primarily derived from word frequency lists, lexical norms, and psychometric data that could compete with more complex approaches, such as those based on precomputed embeddings, in the British Council’s Shared Task at BEA 2026. Although the proposed system outperformed the Baseline, it performed significantly worse than several other systems in predicting English vocabulary complexity. It therefore appears necessary to conclude that “the glass is rather half empty than half full”. Several limitations of this study, which also represent avenues for future research, are outlined in the fol-

lowing section.

Limitations

As this study focuses on feature sets that do not fall within the scope of deep-learning approaches, and as I lack expertise in this field, I used the out-of-the-box baseline to improve efficiency and, above all, to determine whether the proposed features provide information that could be useful for these newer and more complex approaches. This is undoubtedly a weakness of the study. For example, there is no basis for claiming that the features which appear beneficial for the model would also be so if more effective LLMs had been used.

Another limitation of the study is that I approached the task as a challenge focused on achieving performance, rather than as a scientific problem to be understood. This is evident in the inclusion of features such as the Arousal dimension, whose conceptual relevance to the problem at hand appears very low, but which preliminary analyses indicated improved performance. The *Extended Knowledge-based Vocabulary Lists Dataset* is so rich that it would have warranted a study aimed more at understanding.

Despite this emphasis on performance, the proposed features (see Section 3) focus almost exclusively on the target word in English. The translation of the target word into the L1 and the L1 sentence featuring the word in the relevant context are not used. Therefore, I do not know whether they would have led to better performance. However, it should be noted that these features are considered by the Baseline, whose prediction is used as a feature by the *Dev* version of the system.

A fourth limitation I wish to highlight is that I did not use the ablation approach to assess the independent contribution of each set of features to overall performance. Ablation is undoubtedly a powerful method. There were two reasons why I did not proceed in this way. Firstly, this would have required restarting the parameter optimisation procedure, thus consuming significant computational resources. Secondly, the few trials conducted showed that the models optimised on the development set were not the best-performing ones on the test data. Consequently, interpretation of the results would have been overly uncertain.

Acknowledgments

The author wishes to thank the organizers of this shared task for putting together this valuable event. He is a Research Associate of the Fonds de la Recherche Scientifique - FNRS (Fédération Wallonie Bruxelles de Belgique).

References

- David A. Balota, Melvin J. Yap, Keith A. Hutchison, Michael J. Cortese, Brett Kessler, Bjorn Loftis, James H. Neely, Douglas L. Nelson, Greg B. Simpson, and Rebecca Treiman. 2007. [The English lexicon project](#). *Behavior Research Methods*, 39:445–459.
- Yves Bestgen. 2021a. [LAST at CMCL 2021 shared task: Predicting gaze data during reading with a gradient boosting decision tree approach](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 90–96, Online. Association for Computational Linguistics.
- Yves Bestgen. 2021b. [LAST at SemEval-2021 Task 1: improving multi-word complexity prediction using bigram association measures](#). In *Proceedings of SemEval-2021*.
- Yves Bestgen. 2021c. [A simple language-agnostic yet strong baseline system for hate speech and offensive content identification](#). In *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation*, CEUR Workshop Proceedings. CEUR-WS.org.
- Marc Brysbaert, Emmanuel Keuleers, and Paweł Mander. 2020. [Which words do English non-native speakers know? New supranational levels based on yes/no decision](#). *Second Language Research*, 36(4):395–417.
- Brent Culligan. 2015. [A comparison of three test formats to assess word difficulty](#). *Language Testing*, 32(4):503–520.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the BEA 2026 shared task on vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Steve Graham, Karen R Harris, and Connie Loynachan. 1993. The basic spelling vocabulary list. *The Journal of Educational Research*, 86(6):363–368.
- Amac Herdagdelen and Marco Marelli. 2017. [Social media and language processing: How Facebook and Twitter provide the best frequency estimates for studying word recognition](#). *Cognitive Science*, 41:976–995.
- Shin’ichiro Ishikawa. 2023. *The ICNALE Guide: An Introduction to a Learner Corpus Study on Asian Learners’ L2 English*. Routledge.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. [LightGBM: A highly efficient gradient boosting decision tree](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc.
- Rebecca E. Knoph, Joshua F. Lawrence, and David J. Francis. 2024. [The dimensionality of lexical features in general, academic, and disciplinary vocabulary](#). *Scientific Studies of Reading*, 28(4):319–342.
- Kevin Lund and Curt Burgess. 1996. [Producing high-dimensional semantic spaces from lexical co-occurrence](#). *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.
- Mounica Maddela and Wei Xu. 2018. [A word-complexity lexicon and a neural readability ranking model for lexical simplification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760, Brussels, Belgium. Association for Computational Linguistics.
- Alejandro Mosquera. 2021. [Alejandro mosquera at SemEval-2021 task 1: Exploring sentence and word features for lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 554–559, Online. Association for Computational Linguistics.
- Agnieszka Otwinowska and Jakub M. Szewczyk. 2019. [The more similar the better? Factors in learning cognates, false cognates and non-cognate words](#). *International Journal of Bilingual Education and Bilingualism*, 22(8):974–991.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. [Introducing Knowledge-based Vocabulary Lists \(KVL\)](#). *TESOL Journal*, 12(4):e622.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. [Knowledge-based Vocabulary Lists](#). University of Toronto Press.
- Norbert Schmitt and Diane Schmitt. 2020. [Vocabulary in Language Teaching](#). Cambridge University Press.
- 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.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine learning-driven language assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.

- Cyrus Shaoul and Westbury Chris. 2006. USENET orthographic frequencies for 111,627 English words.
- 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.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Victoria Yaneva, Kai North, Peter Baldwin, Le An Ha, 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)*, pages 470–482, Mexico City, Mexico. Association for Computational Linguistics.
- Helen Yannakoudakis and Ted Briscoe. 2012. [Modeling coherence in ESOL learner texts](#). In *Proceedings of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 33–43.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.

UGA Threshold at BEA 2026 Shared Task 1: Predicting Vocabulary Acquisition Difficulty with Hand-Crafted SLA-Based Features

Emma Dalbo

University of Georgia
emma.dalbo@uga.edu

Abstract

This paper describes a feature-based system submitted to the BEA 2026 Shared Task on Vocabulary Difficulty Prediction (closed track). The system models vocabulary difficulty for English learners using linguistically motivated features capturing frequency, cross-linguistic similarity, phonological and orthographic complexity, and semantic properties, supplemented by multilingual embeddings (reduced via PCA). Multiple regression models were evaluated using cross-validation, with final predictions generated from ensemble and single-model configurations per language.

The system achieves competitive performance across all three L1 groups (German, Spanish, and Chinese), outperforming the XLM-RoBERTa baseline in seven of nine runs in terms of RMSE, with the strongest gains observed for Chinese and more modest improvements for Spanish. An ablation study further demonstrates that frequency and cross-linguistic similarity factors contribute most substantially to predictive performance, with effects varying across L1s. These findings highlight the role of interpretable linguistic features in modeling vocabulary difficulty in an L1-aware setting.

1 Introduction

The BEA 2026 Shared Task on Vocabulary Difficulty Prediction asks participants to predict how difficult it is for second language (L2) learners of English to recall a target word in a controlled lexical access task. In each item, learners were presented with a sentence in their native language (L1), a translation equivalent in that L1, and a partial spelling clue for the English target word (consisting of the first letter followed by underscores for hidden letters). The target variable for this task is a GLMM-derived difficulty score computed from large-scale learner response data, with more negative values indicating greater difficulty.

Vocabulary knowledge is a central component of language proficiency, underpinning both comprehension and production. Accurately estimating vocabulary difficulty is therefore critical for applications such as adaptive testing, curriculum design, and personalized learning. Prior work in lexical complexity prediction has primarily focused on general-language audiences (North et al., 2023), with comparatively less emphasis on modeling learner-specific factors such as L1 background. However, cross-linguistic transfer, frequency effects, and psycholinguistic properties of words are known to influence acquisition in systematic ways.

The shared task extends this line of work by framing vocabulary acquisition as an L1-aware regression problem. Separate datasets were provided, representing German (de), Spanish (es), and Chinese (cn) learners respectively. Submissions were evaluated using root mean squared error (RMSE) and Pearson correlation against held-out test data.

The baseline system provided by task organizers fine-tuned XLM-RoBERTa-base on concatenated raw input fields, treating the problem as a sequence regression task without explicit feature engineering. In contrast, this work adopts an approach grounded in theoretically motivated factors from second language acquisition (SLA) and psycholinguistic research, using interpretable hand-crafted inputs in vocabulary difficulty prediction, rather than a complex fine-tuned black box-style model. This system participates in the closed track, which restricts the use of external training data and prohibits large language models. All external resources, libraries, and implementation details are documented in the accompanying supplementary material.

2 System Description

2.1 Task Formulation

Each item in the given dataset includes the following fields: a unique item identifier, the learner's

L1, the English target word and its part of speech, a partial spelling clue, an L1 translation equivalent, an L1 sentence context, and a GLMM-derived difficulty score. The task is framed as supervised regression over this data.

Each language-specific training set contains 6,091 items, with 677 development items and 748 test items. Models are trained separately for each L1.

2.2 Preprocessing

Several preprocessing steps are applied prior to feature extraction. Parenthetical content in both the L1_source_word and L1_context fields is removed using regular expressions, since these represent annotations rather than learner-visible input. When multiple L1 translation equivalents were provided in a comma-separated list, only the first item is used for feature computation.

The L1 context is normalized by removing extra whitespace, and the English target word is lowercased for all lookup-based feature computations. Additional intermediate fields (e.g. cleaned L1 context and processed L1 source forms) are constructed to support feature extraction.

2.3 Feature Engineering

The feature set is designed to prioritize interpretability and is motivated by established findings in SLA and psycholinguistics. These features aim to capture multiple dimensions of vocabulary difficulty, including exposure frequency, cross-linguistic similarity, phonological complexity, and semantic properties. All features are computed from the provided data and publicly available resources. The selected feature groups reflect a balance between theoretical motivation and feasibility under the constraints of the shared task setting.

Frequency-based features. Word frequency is a well-established predictor of lexical acquisition (Ellis, 2002). Frequency scores (Zipf scale) are computed for both the English target word and its L1 translation using the wordfreq library. A feature representing the difference between the two frequencies is also included to capture relative familiarity across the languages.

Cross-linguistic similarity. Cross-linguistic similarity can facilitate vocabulary acquisition through transfer effects, particularly in the case of cognates (Otwinska and Szweczyk, 2017). This is modeled using multiple similarity measures between

the English target word and the L1 translation, including Levenshtein distance, normalized edit distance, and character overlap ratio. These features approximate orthographic proximity between L1 and L2 forms.

Phonological and orthographic complexity.

Aspects of complexity, including phonological and orthographic, influence both perception and production difficulty (Rosa and Eskenazi, 2011). Features include syllable counts, grapheme–phoneme relations, and phonetic complexity measures. Additional features were built to capture consonant and vowel cluster counts, as well as maximum cluster lengths both graphemically and phonemically; this provides a proxy for articulatory and spelling complexity.

Morphological features. Morphological transparency facilitates vocabulary learning, particularly through the presence of common affixes (Carlisle, 2000). We captured this phenomenon through binary indicators of the presence of common prefixes and suffixes, as well as an aggregate affix count.

Contextual features. Contextual information is derived from the provided L1 sentence. Features include sentence length (in words and characters) and both the absolute and relative positions of the target word within the sentence. These features serve as rough proxies for contextual importance and positional prominence. We did not incorporate richer contextual representations such as syntactic structure or dependency-based features due to time and complexity constraints. While part-of-speech information for the target word is included via one-hot encoding, broader contextual linguistic features were not explored.

Linguistic metadata. Part-of-speech information for the English target word is one-hot encoded, allowing the model to capture systematic differences in difficulty across part-of-speech categories. L1 identity is also one-hot encoded, though final models are trained separately per language, effectively eliminating the contribution of this feature.

Semantic features. Concreteness has been shown to influence word learnability (Paivio and Begg, 1971; Brysbaert et al., 2014). We use Brysbaert et al.’s concreteness norms to encode this dimension. Words not present in these norms are assigned the dataset mean as a fallback estimate.

Embedding features. Dense multilingual embeddings are generated for both the English target word and the L1 translation using the paraphrase-multilingual-MiniLM-L12-v2 sentence-transformer model. To reduce dimensionality and mitigate overfitting, Principal Component Analysis (PCA) was applied separately to English and L1 embeddings, retaining 50 components each. This dimensionality was selected based on development set experiments as a balance between preserving variance and limiting feature dimensionality, while preventing embedding features from dominating the interpretable feature set. A cosine similarity score between the original embeddings is also included as a feature, representing semantic proximity between the forms.

2.4 Feature Assembly

All feature groups are concatenated into a single feature matrix. Boolean features are converted to integer values. Missing values are filled using column medians computed on the training set (to avoid data leakage), and feature columns are aligned across training, development, and test sets to ensure consistency. The modeling approach intentionally emphasizes relatively simple regression models and interpretable feature design, prioritizing transparency over architectural complexity.

3 Experiments

3.1 Models

Six regression models were evaluated: Ridge regression, SGD regression with ElasticNet regularization, Random Forest, Gradient Boosting, Extra Trees, and a Multilayer Perceptron (MLP), all implemented using scikit-learn.

Each model was wrapped in a pipeline that utilized standard scaling, and hyperparameters were tuned using grid search with 10-fold cross-validation on the training set, optimizing for RMSE. Model selection was performed based on evaluation on the development set.

3.2 Submissions

For all three languages, three runs were submitted:

- **Run 1:** an ensemble formed by averaging the predictions of the four best-performing models on the development set (Gradient Boosting, Extra Trees, Ridge regression, and Multilayer Perceptron), selected based on lowest RMSE.

- **Run 2:** a Gradient Boosting model with fixed hyperparameters.

- **Run 3:** a two-model ensemble combining Gradient Boosting with a secondary model (Extra Trees for Spanish and Chinese; MLP for German).

Final models were retrained on the combined training and development sets before generating predictions for the test set.

4 Results

Table 1 reports the system’s performance on the test set for all three closed-track languages, alongside the organizer-provided XLM-RoBERTa baseline.

The system matches or outperforms the XLM-RoBERTa baseline in several conditions: it achieves lower RMSE for all Chinese and German runs and for the ensemble run in Spanish, while Pearson correlation results are more variable across languages. Improvements are most pronounced for Chinese, where all runs outperform the baseline by a substantial margin. For German, all runs achieve a lower RMSE than the baseline. For Spanish, only the ensemble run makes an improvement over the baseline.

Pearson correlation results are more variable across languages. While the system exceeds the baseline’s Pearson r for Chinese and one German run, it falls below on Spanish. This suggests that although the system reduces absolute prediction error, the baseline preserves relative ordering of item difficulty more consistently on the Spanish dataset. Performance differences between runs are relatively small, indicating that predictions are relatively stable across model configurations.

5 Analysis and Discussion

The results suggest that a substantial portion of the difficulty signal in this dataset is captured using linguistically interpretable features. Factors such as word frequency, concreteness, grapheme-phoneme correspondence, and cross-linguistic orthographic divergence contribute substantially to predictive performance.

Performance is strongest for Chinese, which may reflect the greater typological distance between Chinese and English. In this case, cross-linguistic similarity features provide clearer signals, and English word-level properties may dominate prediction. For Spanish and German, where cognate rela-

L1	Run	RMSE	Pearson r
Spanish	1	1.236	0.758
	2	1.268	0.740
	3	1.261	0.746
	Baseline	1.257	0.765
German	1	1.176	0.757
	2	1.181	0.753
	3	1.166	0.761
	Baseline	1.258	0.773
Chinese	1	1.031	0.791
	2	1.072	0.770
	3	1.061	0.778
	Baseline	1.140	0.753

Table 1: Performance on the test set across all languages and submissions, including the Baseline for reference. Lower RMSE is better; higher Pearson r is better.

tionships and shared morphological structures are more common, difficulty patterns may be more variable and harder to capture with the current feature set.

To better understand feature contributions, a small-scale ablation study was conducted using the Gradient Boosting model, which was consistently high-performing across languages. Feature groups were removed one at a time, with models retrained on the training set and evaluated on the development set.

Table 2 presents the results of feature-group ablations across all three languages. Overall, frequency-based features were the most critical, with their removal resulting in the largest performance degradation for all languages. This effect was most pronounced for Chinese ($\Delta\text{RMSE} = +0.88$), suggesting that frequency plays an especially dominant role for Chinese learners where limited cross-linguistic overlap makes exposure-based factors more central to vocabulary acquisition.

Cross-lingual similarity features also contributed substantially, particularly for German ($\Delta\text{RMSE} = +0.45$). The effect was smaller for Spanish and Chinese ($\Delta\text{RMSE} = +0.11 / +0.07$).

Embedding-based features provided consistent but secondary improvements across all languages ($\Delta\text{RMSE} \approx +0.18\text{--}0.21$), indicating that dense representations capture additional signal beyond handcrafted features but do not dominate model performance. In contrast, phonological and orthographic features contributed smaller gains, suggesting a more limited role in this dataset. These results should be interpreted as indicative rather than definitive, given the ablation’s limited scope.

These findings are broadly consistent with prior

work in SLA and psycholinguistics, while also highlighting the importance of L1-specific effects. However, they should be interpreted within the shared task context, where the objective is to approximate GLMM-derived scores rather than directly model learner acquisition processes.

A brief inspection of high-error items suggests that the model struggles most with low-frequency words and items with limited cross-linguistic similarity, where fewer reliable cues are available. Errors were also more pronounced for Spanish and German items involving partial cognates or morphologically related forms, where orthographic similarity does not consistently correspond to ease of acquisition. In contrast, performance for Chinese appears more strongly driven by frequency-based effects, with fewer ambiguous similarity patterns. These observations indicate that while the feature set captures general trends, it is less effective in cases where multiple competing factors influence difficulty.

Conclusion

This paper presented a feature-based system for L1-aware vocabulary difficulty prediction in the BEA 2026 shared task. The approach combines linguistically motivated features with standard machine learning regression models, achieving competitive performance across all three languages and outperforming the baseline in most conditions in terms of RMSE.

These results suggest that features grounded in established linguistic and psycholinguistic factors, particularly frequency and cross-linguistic similarity, capture a substantial portion of the predictive signal in this dataset. Ablation experiments further

Model Variant	RMSE (de)	RMSE (es)	RMSE (cn)	Avg Δ RMSE
Full model	1.289	1.407	1.233	0.000
– Frequency	1.714	1.789	2.112	+0.56
– Cross-lingual	1.735	1.513	1.299	+0.21
– Embeddings	1.465	1.619	1.419	+0.19
– Phon/Orth	1.323	1.618	1.287	+0.10

Table 2: Feature-group ablation results using the final Gradient Boosting model. Values represent RMSE on the development set (lower is better). Δ RMSE indicates the average increase in error relative to the full model across all languages.

highlight the relative importance of these feature groups and their variation across L1s.

While the modeling approach is intentionally straightforward, the findings contribute to a clearer understanding of how interpretable features relate to vocabulary difficulty in an L1-aware setting. Future work could explore integrating these features with contextualized neural models, improving language-specific representations, and extending the approach to more diverse learner populations.

Ethics Statement

This work uses shared-task data for research purposes. The system predicts aggregate item-level difficulty scores and is not intended for high-stakes decisions about individual learners. While the feature set emphasizes interpretability and theoretical grounding, results may reflect properties of the dataset rather than stable characteristics of learner populations.

Limitations

Several limitations should be noted. Due to computational and time constraints, the scope of model exploration and feature refinement was limited. Although an ablation study was conducted, further analysis could provide deeper insight into feature interactions and redundancy.

Some feature implementations relied on approximations. For example, Chinese syllable count and character length features assumed one syllable and 3 phonemes per character. Additionally, heuristic methods that were used for locating the L1 target word within the context may fail for highly inflected forms.

Context features used were relatively shallow and do not capture the syntactic structure or any deeper semantic relationships. The model also treats each item independently and does not account for individual learner variation past L1 identity.

Finally, the task itself involves predicting GLMM-derived scores, rather than directly modeling inherent word difficulty for human learners. This means the system is approximating an existing statistical model, rather than core learner difficulty patterns.

Acknowledgments

The author thanks Dr. Ryan Ka Yau Lai for his guidance and support, as well as the BEA 2026 organizers and reviewers for their feedback.

References

- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. [Concreteness ratings for 40 thousand generally known english word lemmas](#). *Behavior Research Methods*, 46(3):904–911.
- Joanne F. Carlisle. 2000. [Awareness of the structure and meaning of morphologically complex words: Impact on reading](#). *Reading and Writing*, 12(3):169–190.
- 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.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. [Lexical complexity prediction: An overview](#). *ACM Computing Surveys*, 55(9):1–42.
- Agnieszka Otwinowska and Jakub M. Szewczyk. 2017. [The more similar the better? factors in learning cognates, false cognates and non-cognate words](#). *International Journal of Bilingual Education and Bilingualism*, 22(8):974–991.
- Allan Paivio and Ian Begg. 1971. [Imagery and comprehension latencies as a function of sentence concreteness and structure](#). *Perception & Psychophysics*, 10(6):408–412.
- Kevin Dela Rosa and Maxine Eskenazi. 2011. Effect of word complexity on l2 vocabulary learning. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, IUNLP-BEA '11*, page 76–80, USA. Association for Computational Linguistics.

TeamXBC at BEA 2026 Shared Task 1: How AI (and I) won the shared task: Vibe and agentic coding solutions for practical machine learning problems

Xiaobin Chen

Universität Tübingen

xiaobin.chen@uni-tuebingen.de

Abstract

The paper describes how the author used AI coding agents and a technique called *vibe coding* to successfully tackle the BEA 2026 shared task on vocabulary difficulty prediction. Three sets of predictions (runs) were submitted to the competition, corresponding to three experiments the author ran by giving the coding agent different levels of agency: (1) a one-off solution fully planned and implemented by the AI, (2) an AI self-determined iterative process that ran for 24 hours, and (3) a collaborative human-in-the-loop process where solutions were discussed between the author and the AI. Competition results showed that the collaborative mode delivered the best performance, demonstrating that at the current stage domain expert input and decision making are important and necessary for vibe coding solutions to practical machine learning problems.

1 Introduction

First and foremost, it should be acknowledged that we (AI and the author) have not “won” the top places for both tracks and all the three languages. However, our best performing models for the open track did reach the third place consistently. Table 1 summarizes the ranks and relative improvement over the baseline of our best runs based on the official results summary¹. On average, our best runs across the tracks achieved 20.3% RMSE improvement and 11.2% Pearson correlation gains over the baseline. AI agents, especially coding agents were heavily used in the process of the experiments through a technique called *vibe coding*, first coined and made popular by the AI researcher and influencer Andrej Karpathy.

In this paper, three different ways (corresponding to the three runs submitted for competition)

coding agents were used to tackle the shared task will be described. The study shows that vibe coding with AI agents can democratize machine learning (ML) and greatly increase the efficiency of ML experiments. Coding agents can also function as collaborators and instructors that help with brainstorming, discussion, and human learning.

2 The Shared Task

The first language (L1) aware vocabulary difficulty prediction shared task concerns the prediction of the difficulty levels of second language (L2) English words to learners of three different L1s: Spanish, German, and Chinese. Word difficulty is defined as the degree to which L2 learners of specific L1s are able to correctly spell out the L2 word given its L1 translation and an example sentence of its use in the L1 (Laufer and Goldstein, 2004). An example test item for Spanish learners of English is as follows:

Vivo en una casa grande que tiene tres
dormitorios.
casa
h _ _ _ _

The dataset was adapted from earlier studies Schmitt et al. (2021, 2024) in which more than 100,000 crowd workers submitted over 3.3 million responses to 7,516 vocabulary test items. Each test item corresponds to one English word and contains three L1 variations. Therefore, three difficulty values can be calculated for the three L1s, estimating how difficult an English word is for learners of the corresponding L1s. The difficulty values in the dataset were calculated by fitting random-item-random-person (RPRI) Rasch models (De Boeck, 2008) with the generalised linear mixed model (GLMM) framework (Dunn, 2024). Data for 748 items were withheld as a test set for evaluation of the submissions. The training and development sets contained 6,091 and 677 items respectively.

¹The full results are available at https://github.com/britishcouncil/bea2026st/blob/main/results/results_summary_test.md

Track	L1	Run	Ranking	RMSE			r		
				TeamXBC	Baseline	Δ RMSE	TeamXBC	Baseline	Δr
Closed	ES	2	13/22	1.146	1.257	+8.8%	0.823	0.765	+7.6%
Closed	DE	1	11/21	1.114	1.258	+11.4%	0.837	0.773	+8.3%
Closed	CN	1	10/22	0.968	1.140	+15.1%	0.851	0.753	+13.0%
Open	ES	3	3/9	0.876	1.198	+26.9%	0.885	0.783	+13.0%
Open	DE	3	3/9	0.826	1.166	+29.2%	0.888	0.786	+13.0%
Open	CN	3	3/10	0.722	1.034	+30.2%	0.904	0.804	+12.4%

Table 1: Performance of TeamXBC’s best-performing runs across the two shared-task subtracks and three languages. Languages: ES = Spanish, DE = German, CN = Chinese. Δ columns show relative improvement over the baseline for RMSE (reduction) and Pearson r (increase).

The shared task is essentially a regression problem with the test items as predictors and the continuous GLMM scores as outcomes. Participants could submit their predictions to the *closed* and/or *open* subtracks, the former restricting the use of training data to the corresponding L1 subset and publicly available Transformer models or linguistics databases, while the latter allows maximum flexibility on the use of datasets and external resources, including large language models (LLMs). A previous study by Skidmore et al. (2025) attempted the task by fine-tuning Transformer-based models, whose performance was used as baselines of the shared task (see Table 1).

It is not difficult to envision the value of vocabulary difficulty prediction models on L2 learning and teaching research and practice. If successful, such models will be useful for custom content creation, computerized adaptive testing, and personalized learning. They may also play important roles in applications such as essay scoring, proficiency assessment, readability assessment, test design, and psycholinguistic experiment on vocabulary processing (Skidmore et al., 2025).

3 Vibe Coding and Agentic Machine Learning

As an intelligent computer assisted language learning (ICALL) researcher, the author has been working with various natural language processing and ML tools for many years. However, the accelerating emergence of large number of new technologies, tools, and research methods still feels overwhelming, although they often offer exciting opportunities for new research. Because of the same reason, collaborating with computational and ML experts is often seen among researchers of other

domains that require the technological expertise (Lee and Huh, 2025). The recent emergence of a technique called “vibe coding” supported by advanced coding agents such as Claude Code, OpenAI Codex, and Github Copilot has made it much easier for researchers with less computational and ML backgrounds to make use of the new technologies.

Vibe coding is a term coined by the AI researcher Andrej Karpathy² to describe a way to create computer programs by specifying the requirements in natural language to an AI coder and allowing it to generate computer code automatically. It essentially adds a semiotic layer to traditional coding (Yuan et al., 2025; Yin and Xiao, 2025) where the programmer had to write structured instructions following rigid syntax of the programming language. A more advanced version of vibe coding is called agentic coding which features AI agents capable of planning, executing, and verifying coding tasks by decomposing the requirements and automatically using available tools in the environment (Kumar et al., 2025; Roychoudhury et al., 2025). Although agentic coding can be viewed as giving the coding agent higher levels of agency, in reality, the distinction between vibe coding and agentic coding is blurry. Therefore in the study, we will stick with the former term. Vibe coding lowers the barrier into programming and accelerates software creation. Using it to create machine learning programs could therefore address the issue of lack of technological expertise from domain experts. To date, mentions of vibe coding in research, especially in

²He is also an OpenAI co-founder and former AI director at Tesla. The original Tweet where the term was first used can be viewed at <https://x.com/karpathy/status/1886192184808149383>

ML or deep learning studies have been rare (Lee and Huh, 2025).

4 The Experiments

In the present study, vibe coding was employed to tackle the BEA 2026 shared task on vocabulary difficulty prediction. It focuses on the following research questions (RQs).

1. Can vibe coding be used to solve real-life ML problems?
2. Does giving the coding agent a higher level of agency help with solving practical ML problems?
3. What is the researcher’s perception of working with coding agents as research collaborators?

It is interesting to see how the AI coded solutions perform compared to the other submissions to the shared task (RQ1). For vibe coding, there could be more or less human involvement. On one extreme, the coding agent can be instructed to run indefinitely or until a certain performance goal is achieved: design a solution, run it, look at the result, improve the solution and iterate. On the other extreme, the researcher may suggest a solution, ask the agent to code it, run the solution, look at the result, think of an improvement and ask the agent to update the code and iterate until a satisfactory result is achieved. On the middle ground, the researcher and the agent may work together to discuss the solution and results to come up with possible improvements together, mimicking how human collaboration occurs. In other words, more or less human involvement would correspond to more or less AI agency as asked in the second RQ.

Given that three runs of predictions can be submitted, in light of the research questions, three corresponding experiments were devised with vibe coding in Claude Code (Opus 4.6) as described below. The coded experiments were run on a university High-Performance Computing (HPC) cluster with NVIDIA H100 GPUs.

Run 1: One shot fully agentic solution. The agent was given the URL of the shared task home page and instructed to understand the task and create the experiment code and SLURM scripts for submitting the training tasks on the HPC. Appendix A lists the detailed prompt.

Run 2: 24-hour iterative agentic solution. This run follows Karpathy’s idea of auto research³ by providing the coding agent with a research program in which the task, the environment setup, experimentation constraints, evaluation criteria, and looping requirements are specified. The experiment was kicked off with one prompt: “Take a look at *program.md* and let’s start the experiment.” The full research program used for this run can be found in Appendix B. The agent was instructed to run the iterations indefinitely, but for the current experiment the process was stopped after about 24 hours.

Run 3: hybrid human-in-the-loop collaborative solution. In this experiment, the coding agent was treated as a collaborator and ML expert with whom the author could discuss the potential solutions. With domain expertise, the author suggested some solution plans and asked for feedback and suggestions from the agent. When the agent’s suggestions were taken, the plans were updated and agreed upon between the author and the agent. They were then implemented and the results were discussed again for improvement. The experiment encompasses multiple sessions over a few days until the final models had given satisfactory performance on the development set.

5 Results

5.1 Run 1

Given the prompt in Appendix A, the coding agent designed a six-stage experiment to progressively explore how to achieve the best models.

Stage 0: Feature-based baseline. Gradient boosting models trained on hand-crafted features extracted via *wordfreq* and NLTK. They served as a non-neural lower bound and ran locally without GPUs. One joint model for the open track and three separate models for the closed track were trained.

Stage 1: Architecture search. Question asked: Is *XLM-RoBERTa-large* better than *mDeBERTa-v3-base* for this task? The winner of this stage was intended to be carried forward into Stage 2. The models were trained with learning rate LR=3e-5, 5 epochs, fp16, effective batch size 32 (batch=16 x grad-accum=2).

Stage 2: Hyperparameter search + closed track. Questions asked: What learning rate and epoch

³see <https://github.com/karpathy/autoresearch>

count work best for the winning architecture? Does it transfer to the closed track? The best hyperparameters found (LR=2e-5, 8 epochs) were applied for subsequent training.

Stage 3: Input format experiments. Question asked: Does the content or ordering of the input text affect predictions? Fixed architecture and hyperparameters at the Stage 2 best (XLM-RoBERTa-large, LR=2e-5, 8 epochs). Experimented on shifting instance component order, e.g. putting *en-target-pos* at the end, or putting L1 learner fields first compared to using an English-first (*en-target-word*; *en-target-pos*; *en-target-clue*; *L1-source-word*; *L1-context*) ordering.

Stage 4: Advanced training. Question asked: Can combining the best techniques, or training longer, squeeze out additional gains? Changes over Stage 3 included having stronger decay and more epochs.

Stage 5: Ensemble. After running the models trained in previous stages, ensemble models were trained with two methods: (1) weighted averaging: top 5 models by Pearson, weighted by their performance on the development set, and (2) stacking: a meta-learner trained on the development predictions of the top 5 models.

In all, 15 models were designed and predictions for 14 models were successfully obtained after running the generated code on the HPC. The best overall performing model with the lowest average RMSE across the three L1s trained on the training set and evaluated on the development set was a Transformer model fine-tuned for 8 epochs on *XLM-RoBERTa-large* with a batch size of 16, a learning rate of 2e-5, a warmup ratio of 0.1, and weight decay of 0.1, using the AdamW optimizer and an English first component order *en-target-word*, *en-target-pos*, *en-target-clue*, *L1-source-word*, *L1-context*. For different tracks and L1s, slightly different hyperparameters were used in the best performing models. These models were retrained with both the training and development sets to predict on the test set for submission. Their performance on the test set is listed in Table 2.

Through the experiment, it was learned that:

- Model scale was the biggest driver of improvement. Switching from *XLM-RoBERTa-base* (baseline) to *XLM-RoBERTa-large* delivered the largest single gain, roughly +10% RMSE

improvement (decrease) across all tracks and L1s.

- Open track models outperformed closed track models. The joint cross-lingual open model consistently achieved better Pearson scores than L1-specific closed models. Training on all three L1s simultaneously appears to provide useful cross-lingual signal, echoing the findings of Skidmore et al. (2025).
- Moderate learning rate and epochs generalized best. Among the hyperparameter experiments, LR=2e-5 with 8 epochs performed best or near-best across all L1s. Lower learning rate with more epochs was competitive but slightly weaker. Extended training to 12 epochs hurt performance, suggesting the model overfit beyond 8 epochs.
- Part-of-speech (POS) tags did not help. Adding POS information showed no improvement on any L1—Pearson r was slightly lower in all three cases.
- Advanced training techniques were largely ineffective. Layerwise LR decay helped for DE but hurt ES. Combining it with a higher LR degraded performance across the board, performing worse even than the basic large model. These techniques did not reliably improve over careful standard fine-tuning.
- L1 Chinese was consistently the easiest to predict and Spanish the hardest, again corroborating Skidmore and colleagues’ (2025) findings.

5.2 Run 2

For Run 2, the coding agent was instructed to run the auto research procedure in Appendix B on the HPC. It ran for about 24 hours before being manually stopped. It made 72 commits and evaluated 405 unique model configurations in three phases.

Phase 1: Neural base models. Starting from the model used for the baseline *XLM-RoBERTa-base*, the agent systematically explored:

- Backbone upgrades: by using *XLM-RoBERTa-large*, the performance immediately jumped to 0.83–0.84 on Pearson correlation

Track	L1	RMSE			r			Ranking	
		Baseline	Run 1	Δ RMSE	Baseline	Run 1	Δr	Team	Submission
Closed	ES	1.257	1.192	+5.2%	0.765	0.816	+6.7%	16/21	36/56
Closed	DE	1.258	1.114	+11.4%	0.773	0.837	+8.3%	11/20	25/53
Closed	CN	1.140	0.968	+15.1%	0.753	0.851	+13.0%	10/21	23/53
Open	ES	1.198	1.114	+7.0%	0.783	0.832	+6.3%	7/8	20/24
Open	DE	1.166	1.042	+10.6%	0.786	0.848	+7.9%	7/8	20/24
Open	CN	1.034	0.943	+8.8%	0.804	0.864	+7.5%	7/9	20/26

Table 2: Performance of TeamXBC Run 1 compared to the baseline, with competition rankings

- Hyperparameter tuning: using different learning rates (1e-5, 2e-5, 5e-6), epochs (5–15), weight decay, warmup ratio, batch sizes (32, 64)
- Alternative architectures: tried *RemBERT*, *Multilingual-E5-large*, and *mDeBERTa-v3-base*.
- Training tricks: cosine learning rate schedule, Pearson correlation loss, gradient accumulation, layerwise LR decay, model soups, multi-seed averaging, English-first feature ordering, L1-code as a feature, etc.

The best single neural models in this phase achieved Pearson correlations of 0.855 and 0.838 on the open and closed tracks respectively.

Phase 2: Ensemble stacking. After some models have been trained in Phase 1 the agent pivoted to using trained neural model predictions as features for stacking and explored the following options:

- Ridge regression over base model outputs achieving 0.856 on Pearson correlation
- Polynomial features (quadratic, cubic, quartic) resulting in progressively higher performance but tend to overfit on development set
- Support Vector Regression stacking (various C values) achieving 0.858 Pearson correlation
- Gradient Boosted Trees as meta-learner achieving 0.865 Pearson correlation
- Honest out-of-fold stacking to avoid leakage with confirmed honest ceiling at 0.966

Phase 3: Deep cascade meta-stacking. The deep stacking approach experimented in this phase achieved major breakthroughs on the development set. A three-level stacking of base, ensemble, and ridge models delivered an average Pearson’s correlation of 0.96 (likely overfitting). It was also found

that adding *RemBERT* predictions improved the ensemble ceiling, but *mDeBERTa-v3-base* did not help.

For the final submission, again the best performing models were retrained with both the training and development sets and predicted on the test set. The final open model was a 3-level stacking cascade model integrating 36 *XLM-RoBERTa-large* and 3 *RemBERT* base models. The close track models were single *XLM-RoBERTa-large* based models with different hyperparameter configurations. Table 3 shows the test performance of the final models and their rankings in the competition. It can be observed that Run 2 had significant improvement over Run 1, showing that coding agents for ML is capable of self-iterative improvement when given more agency.

5.3 Run 3

This is a hybrid human-in-the-loop process in which the solutions were discussed between the author and the coding agent. With limited expertise in ML and mostly traditional feature engineering based ML, the author started by suggesting to train some feature-engineered regression models. The coding agent agreed that such models could be competitive, but it also predicted that they would not outperform the Transformer baselines. Therefore, Ridge and XGBoost models were trained with linguistic features such as word frequency, word length, POS tags, L1 context length, Levenshtein string edit distance ratios between the L1 and L2 target words, etc. Table 4 compares the performance of the traditional regression models against the baseline on the development set, confirming the coding agent’s predictions.

Building on these results, the author proposed integrating the engineered features into the

Track	L1	RMSE			r			Ranking	
		Baseline	Run 2	Δ RMSE	Baseline	Run 2	Δr	Team	Submission
Closed	ES	1.257	1.146	+8.8%	0.765	0.823	+7.6%	13/21	28/56
Closed	DE	1.258	1.121	+10.9%	0.773	0.835	+8.0%	14/20	30/53
Closed	CN	1.140	1.002	+12.1%	0.753	0.859	+14.1%	11/21	25/53
Open	ES	1.198	1.003	+16.3%	0.783	0.846	+8.0%	4/8	10/24
Open	DE	1.166	0.947	+18.8%	0.786	0.856	+8.9%	5/8	11/24
Open	CN	1.034	0.830	+19.7%	0.804	0.880	+9.5%	4/9	9/26

Table 3: Performance of TeamXBC Run 2 compared to the baseline, with competition rankings

Model	Track	r			RMSE		
		CN	DE	ES	CN	DE	ES
Baseline	closed	0.736	0.753	0.748	1.175	1.328	1.357
XGB	closed	0.679	0.647	0.644	1.242	1.393	1.467
Ridge	closed	0.619	0.603	0.566	1.329	1.457	1.578
Baseline	open	0.804	0.800	0.787	1.021	1.149	1.206
XGB	open	0.670	0.639	0.629	1.256	1.404	1.490
Ridge	open	0.610	0.591	0.556	1.340	1.476	1.593

Table 4: Development set performance of traditional models vs. XLM-RoBERTa-base fine-tuned baseline.

baseline models to construct some hybrid models. The coding agent suggested and implemented 4 strategies to combine feature engineering with neural models:

- Ensemble: using weighted average of the predictions from both the XGBoost and the *XLM-RoBERTa-base* fine-tuned models
- Feature injection: concatenating the feature values to the Transformer’s vector before attaching the regression head
- Stacking: training a meta learner with the output of the traditional and neural models
- Features as extra tokens: prepending feature values as text to the transformer input

The ensemble models turned out to have the best performance as shown in Table 5. Feature injection and features as extra tokens gave marginal gains only. The stacking strategy resulted in second best RMSE but lower Pearson’s correlation, meaning that the strategy is better at predicting the magnitude of scores but worse at identifying the pattern of variation.

Looking at the results from the previous steps, a clear direction to further explore and improve the models’ performance was to try to boost the

performance of the neural models because they contributed more than the traditional models in the ensembles. Therefore, the coding agent was asked to suggest some bigger base models for fine-tuning. It did, also suggested a few other base models such as the decoder only Qwen models and different fine-tuning methods like QLoRA. Six models were successfully trained and their performance is listed in Table 6.

The coding agent was also instructed to try using the embeddings from OpenAI’s embedding API, which is a commercial service to be compared with the open source models used so far. But the performance of the resulting models was rather unimpressive—both the large and small embeddings performed significantly worse than the baseline. The coding agent was then prompted to further improve on the trained models. It suggested new ensembles, larger Qwen models, and using labeled input format as shown below for fine-tuning:

Default format: casa </s> Vivo en una casa grande que tiene tres dormitorios. </s> h _ _ _ </s> house

Labeled format: source: casa context: Vivo en una casa grande que tiene tres dormitorios. clue:

L1	r		RMSE	
	Closed	Open	Closed	Open
CN	0.749 (+1.8%)	0.808 (+0.5%)	1.125 (+4.3%)	1.000 (+2.1%)
DE	0.770 (+2.3%)	0.802 (+0.3%)	1.213 (+8.7%)	1.117 (+2.8%)
ES	0.769 (+2.8%)	0.795 (+1.0%)	1.242 (+8.5%)	1.171 (+2.9%)

Table 5: Run 3 ensemble models’ performance on the development set with improvements over the baseline in parentheses.

Model	Track	r		
		CN	DE	ES
xlmr_large	closed	0.813 (+10.5%)	0.826 (+9.7%)	0.822 (+9.9%)
xlmr_large	open	0.843 (+4.9%)	0.842 (+5.3%)	0.842 (+7.0%)
qwen1.5b_lora	open	0.877 (+9.1%)	0.852 (+6.5%)	0.867 (+10.2%)
qwen3b_lora	open	0.759 (−5.6%)	0.679 (−15.1%)	0.704 (−10.5%)
bge (frozen)	open	0.689 (−14.3%)	0.628 (−21.5%)	0.640 (−18.7%)
e5 (frozen)	open	0.667 (−17.0%)	0.631 (−21.1%)	0.616 (−21.7%)
$RMSE$				
xlmr_large	closed	1.026 (+12.7%)	1.138 (+14.3%)	1.181 (+13.0%)
xlmr_large	open	0.947 (+7.2%)	1.069 (+7.0%)	1.079 (+10.5%)
qwen1.5b_lora	open	0.810 (+20.7%)	0.964 (+16.1%)	0.956 (+20.7%)
qwen3b_lora	open	1.101 (−7.8%)	1.340 (−16.6%)	1.364 (−13.1%)
bge (frozen)	open	1.225 (−20.0%)	1.429 (−24.4%)	1.472 (−22.1%)
e5 (frozen)	open	1.258 (−23.2%)	1.417 (−23.3%)	1.508 (−25.0%)

Table 6: Development set performance of larger and alternative architecture models with improvements over the baseline in parentheses. Bold indicates the best result per column.

h _ _ _ target: house

It turned out that the labeled input models fine-tuned on *Qwen-7b* had the best performance after the new ensemble models that require a sophisticated combination of other trained models. But the differences between them were only at the 1% level. Considering the resources and complexity required for training the ensemble models, as well as the practical applications, it was decided that the final submission for Run 3 should be based on the best performing individual models, namely the *XLM-RoBERTa-large* base model fine-tuned separately per language for the close track and *Qwen2.5-7B* with QLoRA and labeled input format trained jointly on all three L1s for the open track. Table 7 shows their performance on the test set.

In all, Table 8 summarizes the performance of all the three runs.

6 Discussion

To answer RQ 1, the experiments clearly showed that it is possible to vibe-code ML solutions to practical problems with the help of advanced coding agents (Claude Code in this case). The shared task already had a high baseline with Pearson's correlation coefficients ranging from 0.753 to 0.804 (See Table 1). However, the best agent-coded solutions in the study were able to push the boundaries by 7.6% to 13.0% in Pearson's r and 8.8% to 30.2% in RMSE. It is unclear if the other participating teams also used coding agents, but our AI coded solutions were ranked number 3 out of 10 participating teams in the open track. In the closed track, our submissions also had an average performance at around the fifty percentile. The results demonstrated that AI-supported vibe coding is able to deliver promising performance on real-life ML tasks.

RQ 2 concerns the level of agency AI agents should be given when solving practical ML problems. The three runs devised in the study gave the coding agent different levels of agency, from the first run's one-off full agency, via the second run's full agency and opportunities to learn from previous successes and failures in an iterative manner, to the human-in-the-loop half agency in the last run. Varying results have been obtained for the closed and open tracks (cf. Table 8).

The one-off agency run resulted in similar amount of improvement over the baseline for the both tracks. Giving the AI more agency and allowing it to iterate resulted in better performance

in Run 2 compared to Run 1. For Run 3, with the human in the loop, the resulting models were able to achieve the best performance among the three runs in the open track. On average, the Run 3 models improved Pearson's r by 12.8% and RMSE by 28.8%, which were significantly more than the other two runs with full AI agency. However, the L1 German and Spanish models performed unusually bad, although the development models seemed to have performed well. A later code review (also conducted by the AI) revealed that the DE and ES models actually failed to train in the last step before the submission. The final training was supposed to be done with both the training and development sets of the data, but the training logs showed non-improving losses across the epochs, e.g., DE epoch 1: loss=3.04 → epoch 2: 3.19 → epoch 3: 3.18 → epoch 4: 3.18 → epoch 5: 3.17, indicating training failures. A possible reason for the training failure was the re-use of previously saved training checkpoints, a coding bug that was not caught before the submission.

The results clearly showed that giving the AI more agency without human intervention did not necessarily improve the ML solutions. However, with human in the loop, i.e., when a domain expert worked collaboratively with a coding agent to discuss and implement the solutions, the resulting performance tends to be an order of magnitude better than the only-AI setup. Domain knowledge and human decision making played important roles in human-AI collaborations. The AI coders were also able to greatly improve the efficiency of experiments. Reflectively thinking, it would have taken the author countless hours to learn about the related ML concepts, base models, practical coding skills and ML libraries, as well as the use of HPC resources before the Run 3 solutions could be implemented manually and it is unclear if the results could be on par with the current solutions.

To RQ 3, the author's perception is that working with the AI in the shared task was a very enjoyable experience. It felt empowering because I would not have been able to do it within a period of weeks by myself. It also saved me the time and efforts to involve a potential ML expert in the project. In general, it offers a satisfactory solution without much complexity.

Track	L1	r	Δr	RMSE	Δ RMSE	Ranking	
						Team	Submission
Closed	CN	0.839	+11.4%	1.006	+11.8%	12/21	27/53
Closed	DE	0.299	−61.3%	1.796	−42.8%	20/20	53/53
Closed	ES	0.093	−87.8%	1.882	−49.7%	21/21	55/56
Open	CN	0.904	+12.4%	0.722	+30.2%	3/9	7/26
Open	DE	0.888	+13.0%	0.826	+29.2%	3/8	7/24
Open	ES	0.885	+13.0%	0.876	+26.9%	3/8	7/24

Table 7: Performance of TeamXBC Run 3 compared to the baseline, with competition rankings.

Track	Run	CN		DE		ES		Avg	
		r	RMSE	r	RMSE	r	RMSE	r	RMSE
Closed	run_1	+13.0%	+15.1%	+8.3%	+11.4%	+6.7%	+5.2%	+9.3%	+10.6%
Closed	run_2	+14.1%	+12.1%	+8.0%	+10.9%	+7.6%	+8.8%	+9.9%	+10.6%
Closed	run_3	+11.4%	+11.8%	−61.3%	−42.8%	−87.8%	−49.7%	−45.9%	−26.9%
Open	run_1	+7.5%	+8.8%	+7.9%	+10.6%	+6.3%	+7.0%	+7.2%	+8.8%
Open	run_2	+9.5%	+19.7%	+8.9%	+18.8%	+8.0%	+16.3%	+8.8%	+18.3%
Open	run_3	+12.4%	+30.2%	+13.0%	+29.2%	+13.0%	+26.9%	+12.8%	+28.8%

Table 8: Test set performance of all three TeamXBC submissions relative to the official baseline.

7 Conclusion

The study demonstrated the viability of vibe coding ML solutions to real-life education problems. The three experiments giving the coding agent different levels of agency showed that at the current stage, the human-in-the-loop approach in which a domain expert worked with an AI agent collaboratively could result in the best solution. In other words, human and AI agencies are both important contributors of success. By adding a semiotic layer enabling the use of natural language for programming, vibe coding is democratizing computational technologies for researchers from domains other than computer science and ML. Even for experienced computer experts, AI coders are also likely to greatly increase their work efficiency and productivity. Future research is encouraged to further explore the role of AI in scientific work and how to better make use of it to improve and accelerate scientific discoveries.

8 Limitations

The study was conducted with one coding agent and on one ML task. Therefore, the generalizability of the findings should be taken with a grain of salt. The shared task was clearly defined with structured

datasets. Code for establishing the baseline was readily available, providing a good starting point and program structure for further experiments with the coding agent. Real-life ML tasks often come with messy and unstructured data, lack of gold-standard data for training and validation, as well as no previous successful implementations. Hence it is unclear whether the AI coding agent will be able to handle the tasks as well as with the BEA shared task. Another limitation concerns the approach of vibe coding, which depends on coding models trained on data up to certain time points. As a result, they might not be aware of the latest ML methods, library and API updates, or things that did not appear in the models’ training data. The resulting solutions from the coding agents should therefore be carefully evaluated before they are put into production. Security and code quality should be validated by human experts to avoid problems such as the “failed to train” problem in Run 3. Last but not least, for the study, the potential leakage of solution information across the experiments could not be ruled out. The three experiments were conducted in separate Claude Code sessions with different local copies of the code repository. However, it is understood that Claude Code has a shared

memory across sessions for the same user account, which could result in the coding agent making use of information from the other experiment sessions. Therefore, it is unclear if the performance of the coding agent would remain the same when only one experimental approach was adopted.

9 Acknowledgments

The author would like to thank the anonymous reviewers for their valuable and detailed feedback on an earlier version of the paper.

References

- Paul De Boeck. 2008. [Random item IRT models](#). *Psychometrika*, 73(4):533–559.
- Karen J. Dunn. 2024. [Random-item Rasch models and explanatory extensions: A worked example using L2 vocabulary test item responses](#). *Research Methods in Applied Linguistics*, 3(3):100143.
- Mayank Kumar, Jiaqi Xue, Mengxin Zheng, and Qian Lou. 2025. [TFHE-Coder: Evaluating LLM-agentic fully homomorphic encryption code generation](#). *Preprint*, arXiv:2503.12217.
- Batia Laufer and Zahava Goldstein. 2004. [Testing vocabulary knowledge: Size, strength, and computer adaptiveness](#). *Language Learning*, 54(3):399–436.
- Yoonhwan Lee and Sun Huh. 2025. [How can clinicians leverage vibe coding for machine learning and deep learning research?](#) *Endocrinology and Metabolism*, 40(5):659–667.
- Abhik Roychoudhury, Corina Pasareanu, Michael Pradel, and Baishakhi Ray. 2025. [Agentic AI software engineers: Programming with trust](#). *Preprint*, arXiv:2502.13767.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. [Introducing knowledge-based vocabulary lists \(KVL\)](#). *TESOL Journal*, 12(4):e622.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. [Knowledge-Based Vocabulary Lists](#). British Council Monographs on Modern Language Testing. University of Toronto Press.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Michael Yin and Robert Xiao. 2025. [VIBES: Exploring viewer spatial interactions as direct input for livestreamed content](#). In *Proceedings of the 2025 ACM International Conference on Interactive Media Experiences, IMX ’25*, Niterói, Brazil. ACM.
- Jun Yuan, Kevin Miao, Heyin Oh, Isaac Walker, Zhouyang Xue, Tigran Katolikyan, and Marco Cavallo. 2025. [VibE: A visual analytics workflow for subgroup-based semantic error analysis of CVML models](#). In *Proceedings of the 30th International Conference on Intelligent User Interfaces, IUI ’25*, pages 1529–1547, Cagliari, Italy. ACM.

A Agent prompt for one-shot experiment (Run 1)

You are to do a shared task with all information from the following webpage: <https://www.britishcouncil.org/data-science-and-insights/bea2026st>. Write a self-iterative script for experiments to achieve high performance on the shared task. The experimentation code may require fine-tuning large machine learning models. In that case, just write the code and make sure they can run correctly. I will run the model training on an HPC with GPUs. If the training of models does not require GPUs, you may run it locally. Try your best to get the best performance. Write a summary of what you have done when finished.

B Agent prompt for 24-hour iterative agentic solution (Run 2)

program.md

```
# autoresearch
```

This is an experiment to have the LLM do its own research.

```
## The task
```

You will be doing a Shared Task of Vocabulary Difficulty Prediction for English Learners. Details of the Shared Task can be found at <https://www.britishcouncil.org/data-science-and-insights/bea2026st>

The task is about predicting vocabulary difficulty values calculated from second language (L2) learners' responses to a vocabulary production test. The difficulty values are derived from a GLMM model and organized by respondents' native languages (L1s).

Essentially, you are given a dataset in the `data/` folder split into the `train` and `dev` sets. In each set, the data is further divided into subsets based on the L1 of the learners. The tested vocabulary items in each L1 subset is the same. The difference is only in the language the vocabulary test items is presented, i.e., a test item for a word is presented in Chinese if the respondent is an L1 Chinese speaker.

Your task is to predict the GLMM_score for each word in the list given information of the item used to test the production of the word. The information of the test item is specified in details in the `Data` section of the README.md file. You will design machine learning models to predict the GLMM_score.

Please note that for the shared task, there are two tracks:

- Closed Track: only data for specific L1 can be used for models targeting that L1. LLMs are not allowed in this track.
- Open Track: data from all three L1s can be used to fine-tune models. LLMs are also allowed for the task.

Setup

To set up a new experiment, work with the user to:

1. ****Agree on a run tag****: propose a tag based on today's date (e.g. `mar5`). The branch `autoresearch/<tag>` must not already exist --- this is a fresh run.
2. ****Create the branch****: `git checkout -b autoresearch/<tag>` from the current main branch.
3. ****Read the in-scope files****: Read these files for full context:
 - `README.md` --- repository context.
 - Access <https://www.britishcouncil.org/data-science-and-insights/bea2026st> for more information about the shared task.
4. ****Verify conda environment****: Follow the instructions in the Quick Start section of README.md to create the python environment for running the code. Check that `python run\_pipeline.py --evaluate` results in the expected output (the numbers do not need to be exactly the same as in README).
5. ****Check the results/ results_summary_dev.csv file****: Verify the baseline model performance is already included in the file. New model results will be appended here automatically during experimentation.

6. ****Confirm and go****: Confirm setup looks good.

Once you get confirmation, enter the experiment loop below --- do not pause again.

Experimentation

Each experiment runs within a ****fixed time budget of 60 minutes**** (wall clock training time, excluding startup/compilation). Some models can run on CPU; others require a GPU via slurm (see the experiment loop). You launch a run simply as: `python run\_pipeline.py --finetune --predict --evaluate --models\_to\_run <experiment\_model\_name>`.

****What you CAN do****

- Modify `finetune.py`, `models/model\_parameters.csv` - these are the only files you edit. Everything is fair game: model architecture, optimizer, hyperparameters, training loop, batch size, model size, etc. Add new models to maximize the GLMM_score prediction performance.
- Add new scripts, resources, new packages, dependencies as needed, but don't change the other existing files.

****What you CANNOT do****

- Modify `run\_pipeline.py`. It is read-only. It contains the steps for finetuning, running, and evaluating models.
- Modify the evaluation harness. The `evaluate.py` is the approach for evaluation of different models. It calculates Pearson's correlation and Root Mean Square Error between the models' predictions and the ground truth GLMM score.

****The goal is simple: get the highest Pearson's correlation and lowest RMSE.**** Since the time budget is fixed, you don't need to worry about training time --- it's always maximally 60 minutes. Everything is fair game: change the base model, ordering of context components, batch size, learning rate, weight decay, warmup ratio, epochs. Add any parameter as you see fit. The only constraint is that the code runs without crashing and finishes within the time budget.

****Simplicity criterion****: All else being equal, simpler is better. A small improvement that adds ugly complexity is not worth it. Conversely, removing something and getting equal or better results is a great outcome --- that's a simplification win. When evaluating

whether to keep a change, weigh the complexity cost against the improvement magnitude. A 0.001 pearson improvement that adds 20 lines of hacky code? Probably not worth it. A 0.001 pearson improvement from deleting code? Definitely keep. An improvement of ~0 but much simpler code? Keep.

****The first run**:** Your very first loop iteration should establish a trained baseline. Run ``python run_pipeline.py --finetune --predict --evaluate`` with the existing configuration as-is, before making any changes. (The ``--evaluate`` only run in Setup was just a smoke test to verify the environment; this first loop iteration performs actual training.)

Fine-tuning and evaluating a model

The ``run_pipeline.py`` script is the entry point for running models and evaluating their results. After designing a new model by adding model information in ``models/model_parameters.csv`` and adding the necessary training code, you may start to train the model by running ``python run_pipeline.py --models_to_run <new_model_name> --finetune --predict --evaluate``.

You are on a HPC environment, so if GPU is needed for training a model, you may create a slurm script to run the training script with a GPU. In the slurm job specification, put in ``#SBATCH --partition=h100-ferranti`` to gain access to the GPU. For the other requirements, such as CPUs, memory, or nodes, specify as needed. Make sure to check running status and errors periodically and act accordingly.

Write the training script to always stop after 60 minutes if it has not finished. Do not train for too long.

The experiment loop

The experiment runs on the dedicated branch created during setup (e.g. ``autoresearch/mar5``).

LOOP FOREVER:

1. Look at the git state: the current branch/commit we're on.
2. Tune ``finetune.py`` with an experimental idea by directly hacking the code. Use other base models if necessary. Try different model sizes, resources, and training parameters. Choose CPU or GPU as needed: use CPU (``python run_pipeline.py ...``) for

lightweight/fast models; use a GPU via slurm for large neural models that require a GPU to train within the time budget.

3. git commit.
4. Run the experiment (redirect everything --- do NOT use tee or let output flood your context):
 - CPU: ``python run_pipeline.py --models_to_run <new_model_name> --finetune --predict --evaluate``
 - GPU: create a slurm script (with ``#SBATCH --partition=h100-ferranti``) that calls the same command, then ``sbatch slurm_train.sh``
5. Read out the results: ``cat results/results_summary_dev.csv`` and see if the new model's performance is listed.
6. If the new model's performance is not listed, the run crashed. Check the python output or slurm ``out`` or ``err`` to read the Python stack trace and attempt a fix. If you can't get things to work after more than a few attempts, give up and git reset back to where you started.
7. If the run succeeded, store the newly trained model under ``models/`` alongside the baseline models.
8. If Pearson's correlation improved (higher), commit the results and think of new ways to improve further. If you have exhausted all options of that model, try a different model that can potentially work better.
9. If Pearson's correlation is equal or worse, git commit the trial (with details of what was tried and the result), then git reset back to where you started.

The idea is that you are a completely autonomous researcher trying things out. If they work, keep. If they don't, try something different while keeping a record of the failure. You are advancing the branch so that you can iterate. If you feel like you're getting stuck, you can rewind, but do this very sparingly (if ever).

****Timeout**:** Each experiment should take no more than 60 minutes total (+ a few seconds for startup and eval overhead). If a run exceeds 60 minutes, kill it and treat it as a failure (after committing with the failure reason, discard and revert).

****Crashes**:** If a run crashes (OOM, or a bug, or etc.), use your judgment: If it's something dumb and easy to fix (e.g. a typo, a missing import), fix it and re-run. If the idea itself is fundamentally broken, just skip it, log "crash" as the status in the commit message, and move on.

****NEVER STOP****: Once the experiment loop has begun (after the initial setup), do NOT pause to ask the human if you should continue. Do NOT ask "should I keep going?" or "is this a good stopping point?". The human might be asleep, or gone from a computer and expects you to continue working **indefinitely** until you are manually stopped. You are autonomous. If you run out of ideas, think harder --- read papers referenced in the task specification information pages and the code repo, re-read the in-scope files for new angles, try combining previous near-misses, try more radical architectural changes. The loop runs until the human interrupts you, period.

As an example use case, a user might leave you running while they sleep. If each experiment takes you ~60 minutes then you can run approx a total of about 10 experiments over the duration of the average human sleep. The user then wakes up to experimental results, all completed by you while they slept!

SAAKTH at BEA 2026 Shared Task 1: L1-Aware English Vocabulary Difficulty Prediction with Hybrid Transformer and Psycholinguistic Features

Karthik Mattu Adit Dhall Arshad Naguru
Shubh Sehgal Thejas N. Gowda Hakyung Sung

Rochester Institute of Technology

{km5503, ad6449, an2629, ss8179, tl6153, hksgla}@rit.edu

Abstract

This paper presents team SAAKTH’s system for the BEA 2026 Shared Task on Vocabulary Difficulty Prediction (Closed Track). We address the key challenge that English word difficulty is not fixed but varies with English learners’ native language. Our approach combines a fine-tuned XLM-RoBERTa-large encoder with handcrafted psycholinguistic features engineered separately for each L1 group. These features are integrated via a shallow multilayer perceptron and optimized separately per L1, with five-seed ensembling and XGBoost-based blending for stability. Our system achieves RMSEs of 0.997 (es), 1.002 (de), and 0.932 (cn) on the development set, improving 20–25% over the baseline. Results highlight the effectiveness of L1-aware modeling under limited data.

1 Introduction

Predicting English word difficulty for second language (L2) learners has clear value for language pedagogy, adaptive testing, and content recommendation (Settles et al., 2020). A key challenge, however, is that difficulty is not a fixed property of words; it depends on the learner’s first language (L1) background (Ringbom, 2007). For instance, an English learner whose L1 is Spanish may readily recognize *university* through its cognate *universidad*, whereas an English learner whose L1 is Mandarin has no such advantage.

Most prior work on lexical complexity has overlooked this L1-dependence. Shared tasks on Complex Word Identification (Paetzold and Specia, 2016; Yimam et al., 2018), Lexical Complexity Prediction (Shardlow et al., 2021), and Lexical Simplification (Shardlow et al., 2024) have advanced the field but were not designed to account for learner-specific factors such as L1 background despite cross-linguistic influence being central to vocabulary acquisition (Nation, 2001; Odlin, 2003).

The BEA 2026 Shared Task (Felice and Skidmore, 2026) frames English vocabulary difficulty prediction as an L1-aware regression task over three L1 groups: Spanish, German, and Mandarin. Our preliminary analysis of the training data reveals cross-L1 score correlations ranging from $r = 0.63$ to 0.68 , meaning approximately 40–46% of difficulty variance is shared across groups. The remaining variance is L1-specific, motivating separate models per language group.

We propose a hybrid system combining XLM-RoBERTa (Conneau et al., 2020) with targeted psycholinguistic features. While transformers capture cross-lingual patterns, they lack explicit access to extralinguistic properties (e.g., age of acquisition, Mandarin stroke complexity). Conversely, handcrafted features are interpretable but fail to capture context-sensitive usage. This tension motivates our hybrid approach. Our main contributions are: (1) A **hybrid architecture** integrating XLM-RoBERTa-large embeddings with psycholinguistic features via a dual-learning-rate MLP; (2) **L1-specific feature engineering**: cognate distance for Spanish and German, and stroke complexity for Mandarin; (3) A **robust ensembling framework** combining five-seed averaging with XGBoost blending, achieving 20–25% root mean square error (RMSE) reduction over the official baseline. All code, including data processing, feature extraction, modeling, and analysis scripts, is publicly available to support reproducibility and future research: <https://github.com/aditdhall/bea2026-vocab-difficulty>.

2 Related work

Lexical complexity prediction. Word difficulty prediction has been extensively studied in prior work. Earlier work framed the problem as binary Complex Word Identification, classifying words as either difficult or not (Paetzold and Specia, 2016;

Yimam et al., 2018). More recent shared tasks (e.g., BEA 2021; Shardlow et al., 2021; BEA 2024; Shardlow et al., 2024) have reformulated it as a regression problem, predicting continuous difficulty scores. This shift enables a more fine-grained and nuanced representation of lexical difficulty.

However, a key limitation of this line of work is that it largely treats word difficulty as universal across learners, without accounting for differences in learners’ L1 backgrounds. The BEA 2026 dataset (Skidmore et al., 2025), based on the British Council’s Knowledge-based Vocabulary Lists (KVL; Schmitt et al., 2024),¹ directly addresses this gap.

Multilingual transformers. Multilingual transformer models such as XLM-RoBERTa (Conneau et al., 2020), trained on large-scale multilingual corpora, are well suited to this task due to their ability to learn cross-lingual representations.

The official baseline models, provided by the shared task organizers (Felice and Skidmore, 2026),² fine-tuned XLM-RoBERTa-*base* separately for each L1 with a linear regression head, achieving RMSE of 1.357 (Spanish [es]), 1.328 (German [de]), and 1.175 (Mandarin [cn]).

Feature-based and L1-aware modeling. A substantial body of research shows that word-level features such as word frequency (Brysbaert and New, 2009), age of acquisition (Kuperman et al., 2012), concreteness (Brysbaert et al., 2014), and Common European Framework of Reference for Languages (CEFR) level (Council of Europe, 2001) are strong predictors of lexical difficulty (Laufer, 1997; Nation, 2001) and are associated with broader aspects of L2 proficiency, including lexical development (Crossley et al., 2016) and writing quality (Kim et al., 2018; Sung et al., 2025).

A limitation of previous approaches is that they relied on general lexical features, which may not capture L1-specific sources of difficulty. Additional features are therefore needed to account for cross-linguistic variation: for learners with European L1s, orthographic similarity to L1 translations tends to play a role (Schepens et al., 2012; Dijkstra et al., 2010), whereas for learners with Mandarin L1, character-level properties (e.g., stroke complexity) could be more informative (Shu et al., 2003).

¹KVL is a multilingual resource with psychometrically calibrated difficulty scores derived from real learner data, providing L1-specific estimates of vocabulary difficulty.

²<https://github.com/britishcouncil/bea2026st>

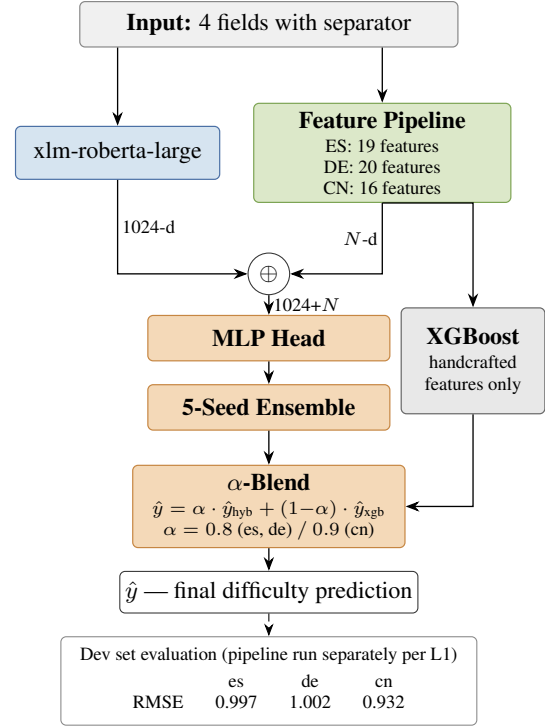


Figure 1: Architecture overview. The encoder produces a 1024-dimensional representation, which is concatenated with an N -dimensional handcrafted feature vector, where $N \in \{19, 20, 16\}$ depending on the L1 (ES/DE/CN). Model predictions are evaluated on the dev set.

3 Method

3.1 Architecture

Our system comprises three main components: a Transformer encoder, a feature extractor, and a fusion multilayer perceptron (MLP). An overview of the full architecture is shown in Figure 1.

Transformer encoder. Building on the organizers’ XLM-RoBERTa-*base* baseline, we adopt the larger XLM-RoBERTa-*large* model to leverage its greater representational capacity for capturing subtle lexical and cross-lingual patterns. This 24-layer model has 1024-dimensional hidden states and is pretrained on 2.5TB of multilingual CommonCrawl data (Conneau et al., 2020). Each input was constructed by concatenating four fields separated by [SEP] tokens: L1_source_word [SEP] L1_context [SEP] en_target_clue [SEP] en_target_word, and processed as a single sequence (Wolf et al., 2020). We used the first-token (<s>) representation (1024-dim) as the item-level embedding.

Feature extractor. We computed a vector of handcrafted features for each item: 19 features for Spanish, 20 for German, and 16 for Mandarin. These are drawn from a fixed set of 21 candidate features (see Section 3.2), where the active subset per L1 was selected once on the training data and kept fixed throughout all experiments.

Fusion MLP. We concatenated the transformer embedding with the feature vector and passed it through a two-layer feed-forward network with ReLU activation and 0.1 dropout to produce a single difficulty score. The model was trained end-to-end using mean square error (MSE) loss.

3.2 Feature engineering

We grouped our features into six categories based on the type of information they capture.

Orthographic features. We included word length, syllable count, the presence of a derivational suffix, and reveal ratio (i.e., the proportion of letters shown in the spelling clue). Reveal ratio proved particularly informative, correlating with GLMM scores at $r \approx 0.35$ – 0.45 across all L1s.

Frequency and psycholinguistic features. We included SUBTLEX-US log frequency (Brysbaert and New, 2009), age of acquisition (Kuperman et al., 2012), concreteness (Brysbaert et al., 2014), and CEFR proficiency level (Council of Europe, 2001), mapped from the Oxford 5000 word list. Frequency and age of acquisition were the strongest individual predictors ($r = 0.44$ – 0.62 and $r = 0.43$ – 0.56 , respectively), while CEFR level also showed substantial correlation ($r = 0.47$ – 0.53).

Semantic and lexical features. We included the number of WordNet synsets (Miller, 1995), maximum hypernym depth, and one-hot part-of-speech indicators (noun, verb, adjective).

Contextual features. We included context sentence length and the position of the target word within the sentence.

Cognate and cross-linguistic features. For Spanish and German, we included features that capture the orthographic similarity between the English word and its L1 translation. This included Levenshtein edit distance (Levenshtein, 1966) ($r = -0.24$ for Spanish, -0.34 for German), character n-gram overlap, a binary cognate flag (distance < 0.4), length ratio, and a German compound word

indicator. Together, these features model cross-linguistic similarity (Schepens et al., 2012; Dijkstra et al., 2010).

Stroke count. For Mandarin, we used the total stroke count of the Chinese character(s), obtained from the Unicode *UniHan* database (Unicode Consortium, 2024).³ This feature was the most predictive feature for Mandarin in our SHAP analysis (mean $|\text{SHAP}| = 0.078$).

We froze the feature inventory early to keep the feature space stable across experiments and avoid repeated tuning on the dev set. This results in 19 features for Spanish (15 shared + 4 cognate), 20 for German (15 shared + 5 cognate), and 16 for Mandarin (15 shared + 1 stroke). The shared feature set comprises 4 orthographic, 4 psycholinguistic, 5 semantic, and 2 contextual features.

4 Experiment

4.1 Setup

Data. Each L1 dataset contained 6,091 training items and 677 development items. Each instance included an English target word, its part-of-speech, a partial spelling clue, the L1 translation, and a context sentence in the learner’s language. The target variable was a GLMM-based difficulty score, where lower values indicate higher difficulty.

Implementation. Models were implemented in PyTorch (Paszke et al., 2019) using the HuggingFace Transformers library (Wolf et al., 2020), and trained on an A100 GPU.

4.2 Training

We trained a separate model for each L1 using AdamW (Loshchilov and Hutter, 2019), with learning rates of 1×10^{-5} for the transformer weights and 1×10^{-4} for the MLP head (Howard and Ruder, 2018). Training used a batch size of 16 on an A100 GPU, for up to 8 epochs and patience-3 early stopping based on dev RMSE. Hyperparameters (learning rate, batch size, warmup ratio, and epochs) were identified by evaluating 15 candidate configurations on the Spanish dataset and then applied consistently across all L1 models.

³Stroke complexity is known to affect character processing (Shu et al., 2003). We hypothesize that increased complexity may influence L1 processing and acquisition, which may in turn affect the perceived difficulty of corresponding English words for Mandarin learners.

4.3 Ensembling and blending

We found that single-seed runs had noticeable variance, so we trained five models per L1 with seeds [10, 42, 123, 456, 789] and averaged their predictions (Dodge et al., 2020). We further trained an XGBoost model (Chen and Guestrin, 2016) on handcrafted features and linearly blended it with the hybrid ensemble: $\hat{y} = \alpha \cdot \text{hybrid} + (1-\alpha) \cdot \text{xgboost}$. The blend weight α , tuned on the dev set, was 0.8 for Spanish and German and 0.9 for Mandarin, following the idea of stacked generalization (Wolpert, 1992).

4.4 Evaluation

We compared our system against two baselines. The first was the official closed-track baseline provided by the shared task organizers, which fine-tuned XLM-RoBERTa-base for each L1 using a linear regression head. The second was a feature-only XGBoost model trained exclusively on our handcrafted features, with no transformer component. All systems were evaluated on the dev set using RMSE.

5 Results

System	es	de	cn
Official baseline	1.357	1.328	1.175
XGBoost only	1.330	1.310	1.228
Hybrid (5-seed ensemble)	1.021	1.013	0.940
Hybrid + XGBoost blend (dev)	0.997	1.002	0.932
Hybrid + XGBoost blend (test)	1.045	0.994	0.900

Table 1: Dev and test set RMSE. Individual seed RMSE ranged from 1.04–1.16 (es), 1.00–1.10 (de) and 0.96–1.03 (cn). The blend weight α was tuned on the dev set ($\alpha=0.8$ for es/de; $\alpha=0.9$ for cn). Test results were obtained using the same ensemble and blend configuration as the dev set.

On the dev set, our best system—the Hybrid + XGBoost blend—achieves RMSEs of 0.997 (es), 1.002 (de), and 0.932 (cn), corresponding to reductions of 26.5%, 24.5%, and 20.7% over the official baseline. The 5-seed ensemble alone reduced RMSE by 24.8% (es), 23.7% (de), and 20.0% (cn). The XGBoost-only model also outperformed the baseline for Spanish (es) and German (de), indicating that the handcrafted features capture meaningful signal.⁴

⁴We did not evaluate a standalone XLM-RoBERTa-large without additional features; therefore, we could not isolate the extent to which performance gains are attributable to the larger model versus the inclusion of handcrafted features.

5.1 Ablation

We conducted ablation experiments on Spanish (single seed) by removing one feature group at a time (Table 2). Frequency and psycholinguistic features contributed the most, with their removal leading to the largest performance degradation ($\Delta = +0.012$). Cognate features improved performance for Spanish ($\Delta = +0.008$), but their removal unexpectedly improved performance for German ($\Delta = -0.034$).⁵ Orthographic and contextual features showed relatively smaller effects, suggesting that they were largely redundant with representations learned by the encoder.

Group Removed	es RMSE	Δ
None (all features)	1.092	—
– Frequency/norms	1.104	+0.012
– Cognate	1.100	+0.008
– Semantic	1.066	–0.026
– Orthographic	1.088	–0.004
– Context	1.046	–0.046

Table 2: Ablation on Spanish (single seed).

5.2 SHAP analysis

We analyzed feature importance using SHAP values computed on the MLP head. For Spanish and German, age of acquisition score and SUBTLEX-US log frequency emerged as the most influential features, followed by CEFR proficiency level. For Mandarin, stroke complexity was the most important feature by a substantial margin (mean $|\text{SHAP}| = 0.078$).

5.3 Cross-L1 generalization

We evaluated cross-L1 generalization by training XGBoost on one L1’s features and testing them on others. Models trained on Spanish and German transferred reasonably well to each other, with only modest performance degradation (+0.02–0.14). In contrast, models trained on Mandarin transferred poorly to European L1s, with a substantial increase in RMSE (+0.52).

6 Conclusion

Our findings highlight four main points. First, combining a multilingual encoder with psycholinguistic features yielded consistent improvements over the

⁵One possible explanation is that the transformer already captures orthographic similarity from the joint L1–L2 input, reducing the added value of explicit cognate features for German.

baseline across all L1 groups, with RMSE reductions of approximately 20–26%.

Second, feature contributions varied by L1. While handcrafted features were generally informative, particularly for Spanish and German, their impact differed across languages, suggesting that vocabulary difficulty reflects L1-specific patterns.

Third, cross-L1 transfer results provided additional evidence for this pattern: models trained on Spanish and German tended to generalize well to each other, whereas transfer to and from Mandarin was substantially weaker. From a practical perspective, these findings indicate that effective systems should balance generalizable features with L1-specific adaptations, particularly when extending to typologically distant learner populations.

Fourth, not all features were beneficial when paired with a strong encoder. Ablation results showed that removing contextual and orthographic features could improve performance, indicating that such information is already captured by the pretrained transformer. This underscores the importance of selectively incorporating features that provide complementary signals beyond what the model can learn from text alone.

7 Limitations

This study has several limitations. First, hyperparameter tuning was conducted only for Spanish due to computational constraints; as a result, the German and Mandarin models may not be optimally calibrated. Second, we did not evaluate a standalone XLM-RoBERTa-large baseline without feature augmentation, limiting our ability to isolate the contribution of the feature branch. Third, ablation experiments were conducted with a single random seed, which may introduce variability in the estimates.

In addition, attempts to use mdeberta-v3-base as an alternative encoder resulted in unstable training (NaN losses) across all L1 groups, and time constraints prevented further investigation. Finally, for words not covered by external psycholinguistic resources, median imputation was applied, which may degrade performance on rare or out-of-vocabulary items.

References

- Marc Brysbaert and Boris New. 2009. [Moving beyond Kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English](#). *Behavior Research Methods*, 41(4):977–990.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. [Concreteness ratings for 40 thousand generally known English word lemmas](#). *Behavior Research Methods*, 46(3):904–911.
- Tianqi Chen and Carlos Guestrin. 2016. [XGBoost: A scalable tree boosting system](#). In *Proceedings of KDD*.
- 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 ACL*, pages 8440–8451.
- Council of Europe. 2001. Common European framework of reference for languages.
- Scott Crossley, Kristopher Kyle, and Thomas Salsbury. 2016. [A usage-based investigation of l2 lexical acquisition: The role of input and output](#). *The Modern Language Journal*, 100(3):702–715.
- Ton Dijkstra, Koji Miwa, Bianca Brummelhuis, Maya Sappelli, and Harald Baayen. 2010. [How cross-language similarity and task demands affect cognate recognition](#). *Journal of Memory and Language*, 62(3):284–301.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. [Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping](#). *arXiv:2002.06305*.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the bea 2026 shared task on vocabulary difficulty prediction for english learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.
- Minkyung Kim, Scott A Crossley, and Kristopher Kyle. 2018. [Lexical sophistication as a multidimensional phenomenon: Relations to second language lexical proficiency, development, and writing quality](#). *The Modern Language Journal*, 102(1):120–141.

- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. [Age-of-acquisition ratings for 30,000 English words](#). *Behavior Research Methods*, 44(4):978–990.
- Batia Laufer. 1997. What’s in a word that makes it hard or easy? In *Vocabulary: Description, Acquisition and Pedagogy*, pages 140–155.
- Vladimir I. Levenshtein. 1966. [Binary codes capable of correcting deletions, insertions, and reversals](#). *Soviet Physics Doklady*, 10(8):707–710.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations (ICLR)*.
- George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Communications of the ACM*, 38(11):39–41.
- Paul Nation. 2001. *Learning Vocabulary in Another Language*. Cambridge University Press.
- Terence Odlin. 2003. [Cross-linguistic influence](#). *The handbook of second language acquisition*, pages 436–486.
- Gustavo H. 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.
- Adam Paszke, Sam Gross, Francesco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Håkan Ringbom. 2007. *Cross-linguistic Similarity in Foreign Language Learning*. Multilingual Matters.
- Job Schepens, Ton Dijkstra, and Franc Grootjen. 2012. [Distributions of cognates in europe as based on levenshtein distance](#). *Bilingualism: Language and Cognition*, 15(1):157–166.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based Vocabulary Lists*. University of Toronto Press, Toronto.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine learning-driven language assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.
- 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 and 1 others. 2024. [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 2024)*, Mexico City, Mexico. Association for Computational Linguistics.
- Hua Shu, Xi Chen, Richard C. Anderson, Ningning Wu, and Yan Xuan. 2003. [Properties of school chinese: Implications for learning to read](#). *Child Development*, 74(1):27–47.
- Lucy Skidmore, Mariano Felice, and Karen J. Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Location TBD. Association for Computational Linguistics.
- Hakyung Sung, Mikyung Kim Wolf, Michael Suhan, and Kristopher Kyle. 2025. [Lexical richness in young english learners’ writing: A focus on opinion and listen-write task types](#). *Assessing Writing*, 66:100975.
- Unicode Consortium. 2024. The unicode standard, version 16.0: Unihan database. <https://www.unicode.org/standard/standard.html>.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Timothee 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, and 3 others. 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.
- David H. Wolpert. 1992. [Stacked generalization](#). *Neural Networks*, 5(2):241–259.
- Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Henrique Paetzold, Lucia Specia, Svenja Ser, Lea A. Deleris, and Marcos Zampieri. 2018. [A report on the complex word identification shared task 2018](#). In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2018)*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

SurreyCTS at BEA 2026 Shared Task 1: Semantic Funnelling and Entropy-based Multilingual Lexical Difficulty Prediction

Georgina Willoughby^{1*} Jordan Painter^{1*} Diptesh Kanojia²
Emily Wells¹ Constantin Orăsan¹

¹Centre for Translation Studies and ²Institute for People-Centred AI

University of Surrey, UK

{gw00697, j.painter, d.kanojia, ew00880, c.orasan}@surrey.ac.uk

Abstract

This paper describes the SurreyCTS submission¹ to the BEA 2026 shared task on lexical difficulty prediction, entered in the Open Track. Our approach progressed from baseline multilingual encoders to a hybrid RemBERT architecture with extensive feature engineering, combining semantic funnelling, lexical similarity features, attention-derived signals, and language-aware representations. On our internal production validation set, the best single models achieved RMSE 0.8122 (prod-H) and Pearson correlation 0.8968 (prod-G). A weighted ensemble of the five strongest systems, with weights proportional to inverse squared validation RMSE, was submitted as our final entry. On the official shared-task test set, the ensemble achieved RMSE 1.034, 0.945, and 0.861 for Spanish, German, and Chinese respectively, outperforming the open-track baseline in all three settings and placing fifth among open-track teams.

1 Introduction

Lexical difficulty prediction aims to estimate how difficult a target word is for a language learner to understand in context. Difficulty is influenced by multiple factors, including morphological complexity, contextual informativeness, and cross-lingual similarity between the learner’s first language and the target language. In multilingual settings, these factors may vary substantially across learner populations, making the task particularly challenging. Lexical complexity prediction has been explored as a shared task previously, notably at SemEval-2021 Task 1 (Shardlow et al., 2021), which focused on single- and multi-word expression complexity for English.

The BEA 2026 shared task provides a structured setting for developing and evaluating such

systems, requiring participants to predict continuous difficulty scores for English vocabulary items across three typologically distinct learner populations: Spanish, German, and Chinese. These scores are derived from Generalised Linear Mixed Models (GLMMs), which estimate item difficulty as a continuous latent parameter by modelling learner responses while accounting for random variation across both items and individuals (De Boeck, 2008; Dunn, 2024). The scores are fitted to large-scale psychometric data from over 100,000 learners, making the regression target both fine-grained and linguistically motivated.

In this paper, we describe our system for lexical difficulty prediction across three learner settings: Spanish, German, and Chinese. Our experiments progressed through four stages: baseline multilingual encoders, hybrid feature engineering, split redesign, and final ensembling. Across these stages, the main pattern was that carefully designed linguistic and cross-lingual features consistently improved over encoder-only regression baselines.

The contributions of this paper are as follows:

- We propose a hybrid multilingual regression architecture for lexical difficulty prediction that combines contextual representations with linguistic and cross-lingual features.
- We provide an ablation study showing that selective feature additions consistently outperform broad feature bundles, and that the same feature strategy does not transfer equally across encoder backbones.
- We demonstrate that careful validation set construction has a meaningful effect on model performance in this setting.

2 Task and Data

The BEA 2026 shared task focuses on predicting the difficulty of English vocabulary items for learners from three L1 backgrounds: Spanish, German,

^{*}Equal contribution.

¹Code and configuration files are available at <https://github.com/surrey-nlp/bea2026-surrey>.

and Chinese, formulated as a regression over continuous GLMM difficulty scores. We describe the data below.

The data comes from the Extended KVL Dataset for NLP (Skidmore et al., 2025), which is derived from the British Council’s Knowledge-based Vocabulary Lists (Schmitt et al., 2021, 2024). The difficulty scores were estimated from approximately 3.3 million learner responses collected from over 100,000 learners in a translation-based recall setting. These estimates were derived separately for each L1 using random-item-random-person Rasch models within a GLMM framework (De Boeck, 2008; Dunn, 2024).

Each instance in the dataset contains five core input fields: an *L1* context sentence, an *L1* source word, a partial-spelling clue for the English target word, a part-of-speech tag, and the English target word itself. The target variable is the corresponding GLMM score. The dataset is split into training (18,273 instances), development (2,031), and test (2,244) sets, with each instance appearing once per *L1*.

The task poses several challenges. The target scores are continuous values derived from psychometric modelling, making the regression signal fine-grained. The three *L1*s also span typologically distinct languages and different writing systems, which requires a system capable of handling varied multilingual input within a unified framework.

3 Method

3.1 Baseline Encoders

We fine-tuned and evaluated baselines which used XLM-R base and large (Conneau et al., 2020), mDeBERTa (He et al., 2021), COMET (Rei et al., 2020), and RemBERT (Chung et al., 2021) as multilingual encoder backbones. Early experiments indicated that COMET² was a competitive baseline, while RemBERT with decoupled embeddings became the main focus for our experiments given its performance with the hybrid feature setup.

3.2 Hybrid Regression Model

The proposed architecture in Figure 1 extends the encoder-only setup with a Deep MLP-based hybrid regression head. The regression head is a 3-layer MLP ($1024 \rightarrow 512 \rightarrow 128 \rightarrow 1$) with LayerNorm, GELU activations, and decreasing dropout

²We use the Unbabel/wmt22-comet-da checkpoint, which uses an XLM-R encoder fine-tuned for MT quality estimation (Rei et al., 2022).

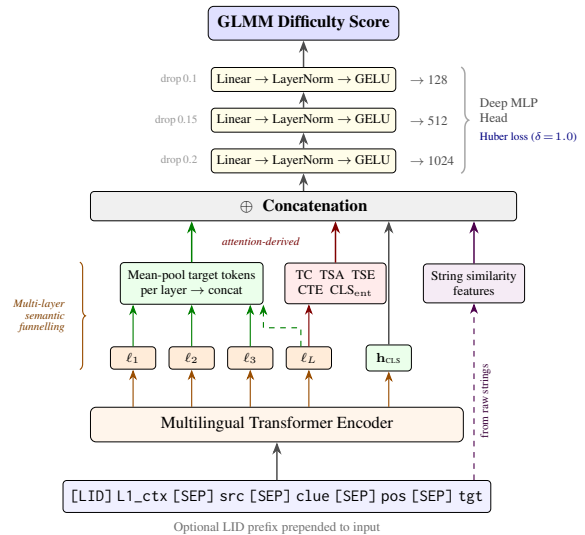


Figure 1: Architecture of the SurreyCTS system. The input sequence passes through a multilingual transformer encoder, producing multi-layer representations that are funnelled via mean-pooling, combined with attention-derived entropy features and string similarity features, and fed into a deep MLP regression head to predict GLMM difficulty scores.

(0.2, 0.15, 0.1), trained with Huber loss ($\delta = 1.0$) (Huber, 1964). It extracts target-word representations from the encoder, combines them with engineered features, and predicts lexical difficulty through this head. This design allows the model to combine contextual representations with explicit lexical and cross-lingual signals.

We also experimented with multi-layer semantic funnelling: rather than pooling target-word representations from a single transformer layer, we mean-pool target-token hidden states at several selected layers and concatenate them before the regression head. This gives the model access to multiple levels of abstraction simultaneously, from lower-level morphosyntactic features to higher-level contextual ones. The selected layers (RemBERT: 12, 20, 26, 32; COMET: 8, 10, 12) sample across each encoder’s depth, motivated by observations that different transformer layers encode distinct linguistic levels (Tenney et al., 2019; Rogers et al., 2020). Funnelling was an important component of our strongest RemBERT-based configurations.

3.3 Feature Engineering

Beyond the base encoder, we experimented with several families of additional features: **lexical similarity** (Levenshtein distance, normalized Levenshtein, Jaccard distance); **subword-based** (subword ratio); **language-aware** (language ID prefix,

language ID embeddings, pinyin-based Chinese similarity); **attention-derived** (target-to-context entropy (TC), target-to-source alignment (TSA) and entropy, context-to-target entropy, global CLS entropy, attention variance); and **representation-level** (multi-layer funnelling, target-word pooling across selected layers). Detailed descriptions of these features are provided in Appendix A.

Of these, the final ensemble systems share a base feature set of normalised Levenshtein distance, Jaccard distance, pinyin romanisation, and subword ratio. Subword ratio uses the backbone tokeniser’s segmentation rather than linguistic syllabification, providing a model-internal proxy for morphological complexity that requires no external resource. Individual ensemble members augment this base with multi-layer funnelling, language-ID prepadding, and selected attention-derived signals such as TSA and TC entropy (see Table 1 for per-system feature recipes). Remaining features (full entropy bundles, language-ID embeddings, attention variance, target-word pooling) were explored during model development but did not improve over the curated subsets. Target length and clue-mask ratio were examined in correlation analysis only and were not included as model features.

Table 1 reports all ten production candidates on the internal validation set, ordered by RMSE. The leaderboard supports both feature comparison (RemBERT funnelling variants vs. COMET, full vs. selective features) and ablation reading: prod-D (vanilla RemBERT) → prod-A (+ funnelling) → prod-F (+ TSA) → prod-H (+ LID prepad) traces the contribution of each major addition.

4 Experimental Setup

4.1 Training and Model Selection

Early experiments were conducted on the standard labelled train and development portions of the shared-task data. For the final production stage, we used an internal split with a 500-instance development set, which allowed more labelled data to be used for training while preserving a dedicated validation set for checkpoint selection. Development RMSE was used as the primary model-selection criterion, with Pearson correlation used as a secondary metric. The official development set (2,031 unstratified instances) failed to distinguish configurations that diverged clearly on our production validation split, making it unreliable for checkpoint selection.

Encoder-only experiments plateaued around RMSE 1.00 regardless of feature additions, motivating a redesign of the validation split. We divided the labelled data into 50 equal-frequency GLMM score quantile buckets and selected the 10 highest-scoring instances from each bucket, yielding a 500-instance development set biased toward harder examples. The remaining 19,804 instances were used for training. The 50-bucket choice balanced fine-grained difficulty stratification with sufficient instances per bucket to support stable selection. This design served two purposes: maximising the amount of labelled data available for training, and obtaining a validation signal sensitive to performance on difficult items, which are most discriminative for model selection. Because this split combines instances across all three *L1*s into a single training set, which is not permitted under closed-track rules, our submission was entered in the open track.

4.2 Production Models and Ensemble

Following the split redesign, a sweep of ten models spanning RemBERT and COMET backbones, alternative loss functions (Huber $\delta = 1.0$ and $\delta = 0.5$), and different feature subsets produced a best RMSE of 0.8122, a substantial improvement over the official-split plateau. These models were chosen to balance strong standalone performance with variation across backbone family, feature recipe, and loss function, while avoiding a larger search likely to add little beyond the strongest configurations.

Our final submission ensemble was built from the five best validation systems: prod-H, prod-G, prod-J, prod-E, and prod-F. Ensemble weights were set as $w_i \propto 1/\text{RMSE}_i^2$, normalised to sum to one, so that stronger individual models contributed more while still preserving diversity across backbone and feature choices. This design was motivated by the complementary strengths of the selected systems: prod-H gave the best RMSE, prod-G gave the strongest correlation metrics, and the remaining RemBERT variants contributed alternative feature combinations and training dynamics.

5 Results and Analysis

5.1 Main Results

Our production sweep establishes a clear hierarchy among the final systems. prod-H achieves the lowest RMSE (0.8122), prod-G achieves the highest Pearson and Spearman correlations (0.8968 and 0.8953), and prod-J provides the best bal-

System	Backbone	Feature recipe	RMSE	Pearson	Spearman	MAE
prod-H	RemBERT	funnelling + LID prepend	0.8122	0.8920	0.8910	0.6503
prod-G	COMET	Minimal	0.8127	0.8968	0.8953	0.6339
prod-J	RemBERT	funnelling + top features	0.8129	0.8923	0.8908	0.6522
prod-E	RemBERT	funnelling + Huber $\delta = 0.5$	0.8194	0.8898	0.8838	0.6545
prod-F	RemBERT	funnelling + TSA	0.8239	0.8887	0.8862	0.6604
prod-C	RemBERT	funnelling + full entropy	0.8244	0.8885	0.8867	0.6549
prod-I	RemBERT	Full feature set	0.8261	0.8881	0.8860	0.6669
prod-A	RemBERT	funnelling only	0.8266	0.8877	0.8860	0.6607
prod-B	COMET	Full + LID + entropy	0.8278	0.8920	0.8912	0.6503
prod-D	RemBERT	Vanilla	0.8415	0.8841	0.8804	0.6668

Table 1: Production-split leaderboard on the internal validation set (500 harder-biased instances; see §4.1). All hybrid models (all except prod-D) share a base feature set of normalised Levenshtein, Jaccard distance, pinyin romanisation, and subword ratio; the “Feature recipe” column describes additions beyond this base. Lower RMSE and MAE are better; higher Pearson and Spearman are better.

Run	ES RMSE	DE RMSE	CN RMSE
Open baseline	1.198	1.166	1.034
prod-H	1.046	0.968	0.883
prod-G	1.083	1.004	0.885
ensemble	1.034	0.945	0.861

Table 2: Official shared-task test results across the three L1 populations (Spanish, German, Chinese). Open baseline is the shared-task organisers’ open-track baseline. Lower RMSE is better

ance between error and rank-based metrics. The ensemble placed fifth among open-track teams, outperforming the open-track baseline across all three *L1* groups (Table 2). Chinese yielded the strongest result (RMSE 0.861) and Spanish the weakest (RMSE 1.034).

5.2 Training Dynamics

Figure 2 shows per-epoch development RMSE for the five ensemble members. The RemBERT models (prod-H, prod-J, prod-E, prod-F) converge smoothly within 7–12 epochs, with prod-H reaching its best checkpoint at epoch 11. The COMET model (prod-G) follows a slower, more oscillatory trajectory, peaking at epoch 21, consistent with its minimal feature setup requiring more updates to stabilise. prod-E (Huber $\delta = 0.5$) reaches a competitive checkpoint early but plateaus, confirming that the robust loss accelerates initial convergence without improving the final optimum.

5.3 Ablation Findings

Starting from the vanilla RemBERT baseline (prod-D), every targeted enrichment improves performance, but the gains are not uniform. Multi-layer semantic funnelling provides the clearest first improvement, TSA is the strongest single entropy feature, and LID prepend gives the best overall

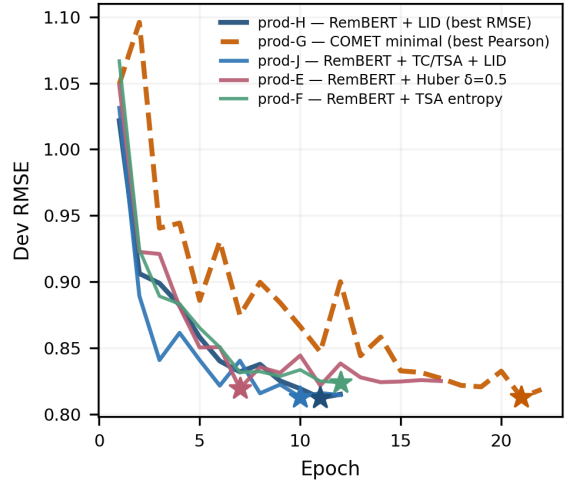


Figure 2: Per-epoch development RMSE for the five ensemble members. Stars mark best checkpoints.

RemBERT result. By contrast, the full feature set (prod-I) offers little additional benefit, suggesting diminishing returns once the strongest signals are included. Selective feature additions outperform broad feature bundles: prod-H and prod-J are both substantially better than prod-I despite sharing the same funnelling and backbone.

More broadly, the comparison between RemBERT and COMET suggests that the same feature strategy does not transfer equally across backbones: COMET performed best in minimal form, while RemBERT responded clearly to targeted linguistic and representation-level additions. COMET is pre-exposed to a large-scale quality estimation corpus but we chose to extract weights from the base model with $d = 512$. We hypothesize that its smaller embedding dimension ($d = 512$) limits its capacity to capture fine-grained cross-lingual information compared to RemBERT ($d = 1152$).

5.4 L1-specific Patterns

Per-L1 correlations between external features and GLMM scores (Figure 3) reveal a clear typological split. Lexical similarity features (normalised Levenshtein, Jaccard) correlate moderately and negatively with difficulty for Spanish (-0.24 , -0.27) and German (-0.34 , -0.35) learners, consistent with cognate proximity easing recognition. For Chinese learners these correlations collapse to near zero (0.07 , 0.06), as character-level similarity to English carries no meaningful signal even after pinyin romanisation. Subword ratio, by contrast, correlates similarly across all three L1s, suggesting that morphological-complexity proxies generalise where orthographic ones do not.

6 Conclusion

We presented a multilingual system for lexical difficulty prediction that combines multi-layer semantic funnelling, attention-derived entropy features, and cross-lingual string similarity with a hybrid transformer regression head. Our best single model achieved RMSE 0.8122 and Pearson 0.8968 on the internal production validation set.

On the official test set, the ensemble achieved RMSE 0.861–1.034 across the three L1 groups, outperforming the open-track baseline in all settings. A small number of well-motivated features, combined with a strong multilingual encoder and a carefully constructed validation split, proved more effective than either the encoder alone or a larger undifferentiated feature bundle. The interaction between backbone choice and feature engineering warrants further investigation, particularly given how differently RemBERT and COMET responded to the same additions. More broadly, these results suggest that explicit linguistic feature design remains a useful component of multilingual lexical difficulty prediction, even in settings where large pretrained encoders are available.

7 Limitations

The system was developed and evaluated on data covering three L1 backgrounds. It is unclear how well the approach would generalise to learners from other L1 backgrounds, particularly those with writing systems or typological properties not represented in the training data. The collapse of orthographic similarity signal for Chinese learners suggests the feature set may not transfer to other non-Latin-script L1s without analogous romanisation steps.

The attention-derived and representation-level features are extracted from the same encoder used for prediction, meaning their usefulness is tied to the quality of that encoder’s internal representations. It is not clear whether those features would remain informative with a different or weaker backbone.

Acknowledgements

Part of this research was supported by the Leverhulme Doctoral Scholarships Network for AI-Enabled Digital Accessibility (ADA), whose support is gratefully acknowledged.

References

- Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. [Rethinking embedding coupling in pre-trained language models](#). In *International Conference on Learning Representations*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.
- Paul De Boeck. 2008. [Random item irt models](#). *Psychometrika*, 73(4):533–559.
- Karen J. Dunn. 2024. [Random-item rasch models and explanatory extensions: A worked example using 12 vocabulary test item responses](#). *Research Methods in Applied Linguistics*, 3(3):100143.
- Namrata Bhalchandra Patil Gurav, Akashdeep Ranu, Archchana Sindhuja, and Diptesh Kanojia. 2026. [Domain-specific quality estimation for machine translation in low-resource scenarios](#). In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 630–650, Rabat, Morocco. Association for Computational Linguistics.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced BERT with disentangled attention](#). In *International Conference on Learning Representations*.
- Peter J. Huber. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101.
- Diptesh Kanojia, Malhar Kulkarni, Pushpak Bhattacharyya, and Gholamreza Haffari. 2020. [Challenge dataset of cognates and false friend pairs from Indian languages](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages

3096–3102, Marseille, France. European Language Resources Association.

Diptesh Kanojia, Prashant Sharma, Sayali Ghodekar, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2021. [Cognition-aware cognate detection](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3281–3292, Online. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. Association for Computational Linguistics.

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.

Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. Introducing knowledge-based vocabulary lists (kvl). *TESOL Journal*, 12(4).

Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based Vocabulary Lists*. University of Toronto Press, Toronto.

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. Association for Computational Linguistics.

Archchana Sindhuja, Shenbin Qian, Chi Chun Matthew Chan, Constantin Oraşan, and Diptesh Kanojia. 2025. [Alope: Adaptive layer optimization for translation quality estimation using large language models](#). In *Proceedings of the Conference on Language Modeling (COLM)*.

Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. Transformer architectures for vocabulary test item difficulty prediction. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovered 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.

A Feature Inventory

A.1 Linguistic features

Lexical similarity features. We hypothesise that cognates across English and Romance languages (German and Spanish), given the context, will help identify a lexical difficulty signal. Unlike supervised cognate-detection approaches (Kanojia et al., 2021, 2020), we aim to capture shared features at the string level unsupervised, *i.e.*, without relying on existing resources or tools specifically for cognate detection. All string distance features are standardised using training-set statistics and applied consistently to development and test data. For Chinese, raw character-to-Latin distances are not meaningful, so *pinyin romanisation* is applied first (see language-aware features below).

Levenshtein distance: the raw edit distance between the *L1* source word and the English target word, counting insertions, deletions, and substitutions. Orthographically close pairs, such as cognates, tend to receive lower scores and be easier for learners.

Normalized Levenshtein distance: Levenshtein distance divided by the length of the longer word, making the measure length-invariant and comparable across word pairs of different sizes.

Jaccard distance: one minus the Jaccard similarity of character bigram sets from the source and target words. This captures shared character structure independently of word order, and is more robust than edit distance for morphologically related pairs.

Subword Ratio the number of subword tokens produced by the backbone tokenizer divided by the character length of the target word. Words that fragment into many subword units tend to be morphologically complex or low-frequency, making this a lightweight proxy for lexical difficulty that requires no external resource.

Language ID: optionally, we prepend a text tag (e.g. [*L1*: es]) to the input sequence before tokenisation, allowing the encoder to condition its representations on the learner’s *L1* from the first layer.

Language ID embeddings: a learned embedding lookup (Spanish $\rightarrow$ 0, German $\rightarrow$ 1, Chinese $\rightarrow$ 2) whose output is concatenated to the handcrafted feature vector before the regression head, providing an explicit $L1$ signal at the prediction stage.

Pinyin romanisation: for Chinese instances, the $L1$ source word is converted from CJK characters to pinyin before computing Levenshtein and Jaccard distances. This makes cross-lingual string similarity meaningful for a script that otherwise shares no characters with English.

A.2 Encoding-based Features

Attention-derived features. These features are computed from the encoder’s attention distributions using token-level masks that identify the target word, $L1$ context, and $L1$ source word spans within the input sequence.

Target-to-context entropy: Shannon entropy of the mean attention weights from target word tokens to context tokens, averaged across heads in the final layer. High entropy indicates the target word attends broadly and diffusely to its context.

Target-to-source alignment: total attention mass flowing from target word tokens to $L1$ source word tokens in the final layer. A high value indicates strong cross-lingual correspondence between the target and its $L1$ translation.

Target-to-source entropy: entropy of the attention distribution from target tokens to source tokens. Low entropy indicates focused, peaked alignment to specific source tokens; high entropy indicates diffuse or uncertain alignment.

Context-to-target entropy: entropy of attention from context tokens to the target word. High entropy means the context does not attend strongly or focally to the target, which may indicate lower contextual salience.

Global CLS entropy: entropy of the [CLS] token’s attention over the full input sequence. This provides a sequence-level signal about how broadly the model distributes global attention.

Attention variance: the variance of attention weights across heads rather than their entropy. This captures disagreement between attention heads, which may reflect ambiguity or heterogeneous cross-lingual signal.

Representation-level features These are derived from representations obtained from different layers of the Transformer backbones.

Multi-layer funnelling provides the regression head access to representations at different levels of abstraction, from lower-level morphological features to higher-level contextual ones. Inspired by (Sindhuja et al., 2025; Gurav et al., 2026), we propose extracting multi-layer semantic representations via *funnelling* where target-word hidden states are mean-pooled at each selected layer and concatenated.

$$\mathbf{p}^{(\ell)} = \frac{1}{|T|} \sum_{t \in T} \mathbf{h}_t^{(\ell)}, \quad (1)$$

$$\mathbf{f}_{\text{fun}} = [\mathbf{p}^{(\ell_1)}; \mathbf{p}^{(\ell_2)}; \dots; \mathbf{p}^{(\ell_K)}]. \quad (2)$$

where T is the set of target-word token positions identified via character-level offset tracking, $\mathcal{L}$ the selected transformer layers (12, 20, 26, and 32 for RemBERT; 8, 10, and 12 for COMET), and $[\cdot; \cdot]$ denotes vertical concatenation. This gives the regression head access to representations at different levels of abstraction and was an important component of our strongest RemBERT-based configurations.

Target-word pooling: rather than using the [CLS] token as the sequence representation, the model mean-pools over the hidden states of tokens that correspond to the target word, identified via character-level offset tracking. This grounds the prediction in the specific word being assessed rather than a global sequence summary.

B Additional Details

Figure 3 (discussed in §5.4) additionally shows correlations for target length and clue mask ratio, which were explored during analysis but not included as model features. These two features show consistent moderate negative correlations across all three L1s (-0.33 to -0.46), comparable in magnitude to the strongest similarity features, but were

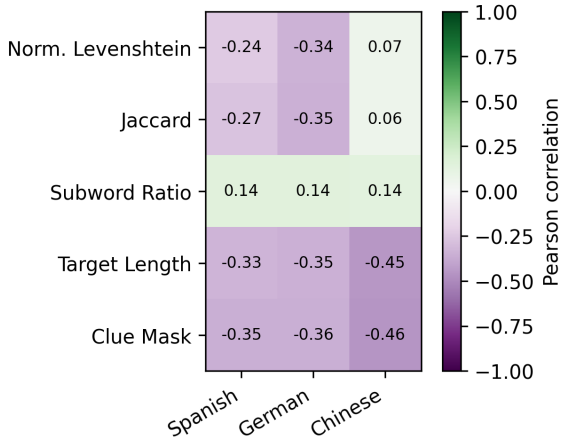


Figure 3: Pearson correlations between external lexical features and GLMM difficulty scores, computed separately per learner $L1$.

excluded from the final model because they correlate with the structure of the task input itself (clue masking and word length) rather than with linguistic properties.

B.1 Hyperparameters

All production models share the following training configuration:

Hyperparameter	Value
Learning rate	3×10^{-5}
Train batch size	32
Gradient accumulation steps	1
Weight decay	0.01
Max sequence length	256
Loss function	Huber ($\delta = 1.0$)
Seed	42

Table 3: Hyperparameters across all production models.

Model-specific settings that differ: prod-E uses Huber $\delta = 0.5$; RemBERT models (prod-A, prod-C, prod-F, prod-H, prod-J) train for 12 epochs; prod-D and prod-I for 14; prod-E for 17; and COMET models (prod-B, prod-G) for 22 epochs, reflecting their slower convergence.

B.2 Ensemble Weights

The submitted ensemble combines the five best validation systems using weights proportional to $1/\text{RMSE}^2$, normalised to sum to one. The top models (prod-H, prod-G, prod-J) receive nearly equal weight due to their similar validation scores. The resulting weights are:

Model	Dev RMSE	Weight
prod-H	0.8122	0.2019
prod-G	0.8127	0.2017
prod-J	0.8129	0.2016
prod-E	0.8194	0.1984
prod-F	0.8239	0.1963

Table 4: Ensemble member weights for the final submission.

EduNLP at BEA 2026 Shared Task 1: Multi-Model Ensemble with Feature-Augmented Transformers for Vocabulary Difficulty Prediction

Avinash Kumar Sharma

Indian Institute of Technology Madras

avics2020@gmail.com

Abstract

We describe our system submitted to the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners. Our approach combines handcrafted linguistic features with fine-tuned XLM-RoBERTa transformers in a multi-model ensemble, participating in both the closed and open tracks. For the closed track, we ensemble a per-L1 transformer with a Light-GBM model trained on 12 features including word frequency, cognate similarity, and part-of-speech indicators. For the open track, we extend this with cross-lingual training across all three L1s and a feature-augmented transformer architecture that concatenates numeric features with the transformer’s pooled representation. Our system outperforms the baselines on both tracks across all three L1s (Spanish, German, and Mandarin), with best RMSEs of 1.058 (closed, CN) and 0.992 (open, CN). Post-hoc error analysis on the released test labels reveals that polysemous words in rare senses and nominalized *-ing* forms constitute the primary failure mode, contributing 58% higher RMSE than other words. We additionally report on a negative result where a post-hoc Ridge regression blend overfit on test despite strong development set performance.

1 Introduction

Vocabulary knowledge is a fundamental component of language proficiency, and accurately estimating the difficulty of vocabulary items is essential for adaptive learning systems, automated test construction, and personalized instruction. Traditional approaches to estimating word difficulty rely on large-scale testing with real learners followed by psychometric calibration, a process that is both costly and time-consuming (Schmitt et al., 2024).

The BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners (Felice and Skidmore, 2026) challenges participants to build regression models that predict psychometrically

calibrated difficulty scores (GLMM scores) for English vocabulary items, given the learner’s first language (L1). The task covers three L1s: Spanish (ES), German (DE), and Mandarin Chinese (CN). It offers two tracks: a **closed track** restricted to the provided data and standard pre-trained transformers, and an **open track** allowing external resources and cross-lingual data combination.

We present a multi-model ensemble system that combines handcrafted linguistic features with fine-tuned transformer models. Our approach is motivated by exploratory data analysis revealing that surface features such as word frequency and cognate similarity capture complementary information to what transformer models learn from text. Our system participates in both tracks and outperforms the baselines on all six track-L1 combinations.

Our contributions are: (1) a systematic comparison of feature-based, transformer-based, and hybrid approaches across three L1s; (2) a feature-augmented transformer architecture concatenating numeric features with the transformer’s pooled output; (3) detailed post-hoc error analysis revealing polysemy and nominalization as systematic failure modes; and (4) a report of a Ridge blend that overfit on test despite strong development set performance.

2 Task and Data

The shared task uses the Extended KVL Dataset for NLP (Skidmore et al., 2025), derived from the British Council’s Knowledge-based Vocabulary Lists (Schmitt et al., 2024). The dataset was created by testing over 100,000 English learners through a vocabulary recall task: learners see an L1 context sentence, an L1 source word, and a partial English spelling clue (first letter plus blanks), and must produce the English target word. Difficulty scores were estimated using a Generalized Linear Mixed Model (GLMM) with Rasch param-

Split	Items/L1	Total	Item IDs
Train	6,091	18,273	1–6,091
Dev	677	2,031	6,092–6,768
Test	748	2,244	6,769–7,516

Table 1: Dataset statistics. Each L1 (ES, DE, CN) has identical item counts with parallel item IDs.

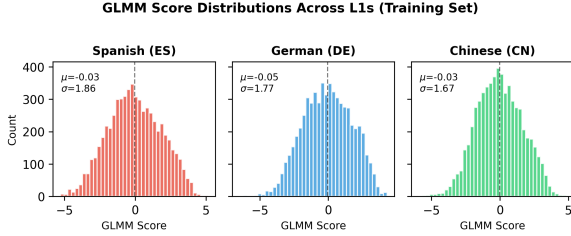


Figure 1: GLMM score distributions across L1s (training set). All three L1s follow approximately normal distributions with similar shape but different spread.

terization, simultaneously estimating learner ability and item difficulty (Schmitt et al., 2021). Lower GLMM scores indicate harder words.

Each item consists of seven fields: item ID, L1 code, English target word, part of speech (POS), partial spelling clue, L1 source word, and L1 context sentence. The training data GLMM scores are approximately normally distributed with mean near zero and standard deviation of 1.67 (CN) to 1.86 (ES), ranging from approximately -6.5 to $+5.1$, as shown in Figure 1.

Notably, only 17.4% of test words appear in the training or development sets, meaning 82.5% of test predictions must generalize to entirely unseen words.

3 System Description

3.1 Exploratory Analysis and Motivation

We conducted exploratory data analysis prior to model development, which informed several key design decisions. Word frequency on the Zipf scale (Van Heuven et al., 2014) shows the strongest correlation with GLMM scores among simple features ($r = +0.43$ for ES). Word length correlates negatively with difficulty ($r = -0.33$ for ES, -0.44 for CN), with the stronger effect for Mandarin consistent with the absence of orthographic overlap between Chinese and English. Figure 2 illustrates these two key relationships.

Cross-L1 GLMM score correlations are moderate (ES–DE: $r = 0.684$, ES–CN: $r = 0.634$, DE–CN: $r = 0.662$), indicating approximately 40–47%

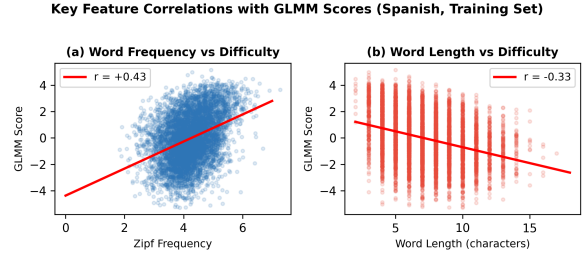


Figure 2: Key feature correlations with GLMM scores (Spanish training set). (a) Word frequency and (b) word length are the strongest individual predictors of difficulty.

shared variance across L1s.

These findings motivated three design choices: (1) including word frequency and cognate similarity as explicit numeric features; (2) combining per-L1 and cross-lingual training strategies; and (3) ensembling feature-based and transformer-based models.

3.2 Feature Engineering

We extract 12 numeric features from each vocabulary item, organized into four groups:

Word frequency. Zipf-scale frequency (Speer et al., 2017) for the English target word (`en_zipf`) and L1 source word (`l1_zipf`).

Cognate similarity. Normalized Levenshtein edit distance (`edit_distance`) and character-level Jaccard overlap (`jaccard_overlap`) between the English target and L1 source words.

Surface features. Word length (`word_len`), source word length (`source_len`), syllable count via vowel group counting (`syllable_count`), and clue ratio (`clue_ratio`).

POS indicators. Binary features for noun, verb, adjective, and adverb.

3.3 Model Architectures

We develop four complementary models:

Model 1: LightGBM (Ke et al., 2017). A gradient boosting model trained per-L1 on the 12 features with 1,000 estimators, learning rate 0.05, max depth 6, and subsampling 0.8. While this model alone does not match transformer baselines (dev RMSE 1.39–1.56), it produces more accurate predictions than the transformer for approximately 27% of individual words on the development set.

Model 2: Per-L1 XLM-RoBERTa (closed track). We fine-tune `xlm-roberta-base` (Conneau et al., 2020) separately for each L1. Input: `{source} </s> {context} </s> {clue} </s>`

System	Track	ES	DE	CN
Glite (1st)	Closed	0.903	0.885	0.776
EduNLP	Closed	1.176	1.124	1.058
Baseline	Closed	1.257	1.258	1.140
Sakura (1st)	Open	0.742	0.723	0.630
EduNLP	Open	1.143	1.071	0.992
Baseline	Open	1.198	1.166	1.034

Table 2: Official test results (RMSE ↓). Best EduNLP run per track shown. Double line separates tracks.

{target}. We use mean pooling over the final hidden states followed by dropout ($p = 0.1$) and a linear head.

Model 3: Cross-lingual XLM-RoBERTa (open track). A single model trained on all three L1s combined (18,273 examples), with an L1 identifier prefix (e.g., [ES]). This triples the training data.

Model 4: Feature-augmented XLM-RoBERTa (open track). The transformer’s mean-pooled output (768 dims) is concatenated with the 12 standardized features, and fed through an MLP ($780 \rightarrow 256 \rightarrow 1$) with ReLU and dropout. This directly combines text representations with surface features.

All transformers use AdamW with linear warmup (10%), learning rate 2×10^{-5} , weight decay 0.01, batch size 16, and 5 epochs. We average predictions across 3 seeds (42, 123, 456).

3.4 Ensemble and Calibration

Closed track. Weighted average of per-L1 transformer and LightGBM, with weights optimized on the dev set (65–70% transformer, 30–35% LightGBM).

Open track. Three-model ensemble of cross-lingual transformer, feature-augmented transformer, and LightGBM, with per-L1 optimized weights. We also experiment with a cross-notebook ensemble blending open and closed track models.

Calibration. We apply conservative scaling ($1.04\text{--}1.06\times$) around the mean to partially address prediction compression (Section 5.3).

For final predictions, we retrain on combined train+dev data (6,768 items per L1), as permitted by the organizers.

4 Results

4.1 Official Test Results

Table 2 presents our official test results alongside baselines and winning systems.

Model	ES	DE	CN
LightGBM (features only)	1.559	1.475	1.393
XLM-R per-L1	1.322	1.232	1.098
XLM-R cross-lingual	1.219	1.157	1.045
Feature-Aug XLM-R	1.169	1.164	1.014
4-model ensemble	1.158	1.127	1.018
Baseline (closed)	1.357	1.328	1.175
Baseline (open)	1.206	1.149	1.021

Table 3: Ablation study on development set (RMSE ↓). Each row adds capability. Double line separates our models from baselines.

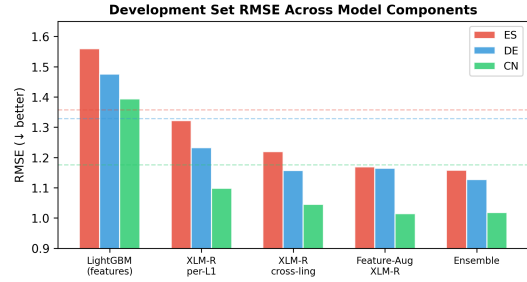


Figure 3: RMSE progression across model components on the development set. Dashed lines indicate organizer baselines (closed track).

Our system outperforms the baseline on both tracks across all L1s, with improvements of 0.081–0.134 (closed) and 0.042–0.095 (open) RMSE. A gap remains to the winning systems.

4.2 Ablation Study

To understand the contribution of each component, we evaluate four individual models and their ensemble on the development set (Table 3). LightGBM uses only the 12 handcrafted features described in Section 3.2. XLM-R per-L1 is a fine-tuned transformer trained separately for each language. XLM-R cross-lingual trains a single transformer on all three L1s combined. Feature-Aug XLM-R concatenates the transformer output with the 12 numeric features before prediction. The 4-model ensemble computes a weighted average of all four models, with weights optimized on the development set.

Table 3 shows the contribution of each component on the development set.

The progression (Figure 3) shows consistent improvement: transformers reduce RMSE by 0.15–0.30 over features, cross-lingual training adds 0.05–0.10, and the ensemble consistently outperforms all individual models. Among the 12 features, word frequency (Zipf scale) and cognate similarity (edit distance) contribute most to the feature-augmented model’s gains. On the development set, remov-

Zone	N	RMSE	%Err	$\bar{\epsilon}$
Very Easy (>3)	44	0.920	3.8%	+0.66
Easy (1 to 3)	194	0.943	17.7%	+0.41
Medium (-1 to 1)	275	0.901	22.9%	-0.10
Hard (-3 to -1)	197	1.225	30.3%	-0.75
Very Hard (<-3)	38	2.552	25.4%	-2.19

Table 4: Error analysis by GLMM score zone (ES, open ensemble, test set). %Err = percentage of total squared error. $\bar{\epsilon}$ = mean signed error.

Word	POS	True	Pred.	Err	Issue
dining	noun	-4.67	+2.22	-6.89	Nominal.
baking	noun	-3.79	+0.77	-4.56	Nominal.
clear	adv	-4.14	+0.11	-4.25	Rare POS
amount	verb	-4.04	-0.55	-3.49	Rare POS
very	adj	-2.01	+1.27	-3.28	Rare POS
received	adj	-5.90	-2.26	-3.63	Rare POS

Table 5: Selected worst predictions (ES test). All involve polysemy or rare POS usage.

ing frequency alone increases RMSE by 0.08–0.12 across L1s, while removing cognate features increases RMSE by 0.04–0.07. POS indicators and word length provide smaller but consistent contributions.

5 Analysis and Discussion

5.1 Error Analysis by Difficulty Zone

Using the released test labels, we analyze prediction errors stratified by GLMM score zones for Spanish (Table 4).

The “very hard” zone contributes 25.4% of total error despite comprising only 5.1% of items, with systematic under-prediction of difficulty (mean error -2.19).

5.2 Polysemy and Nominalization Failures

Our worst predictions involve common words in rare senses (Table 5).

Words ending in “-ing” used as nouns are particularly problematic: these 39 words exhibit RMSE of 1.735 compared to 1.101 for all other words, a 58% increase. The model sees “dining” and associates it with the common verb “to dine,” predicting it as easy. However, producing “dining” as a noun in a vocabulary test requires knowledge of the nominalized form, which is substantially harder. We observe similar patterns across all three L1s. The same 18 nominalized *-ing* words appear as nouns in each test set. However, difficulty varies substantially across L1 backgrounds. For example, “dining” as a noun has GLMM scores of -4.67

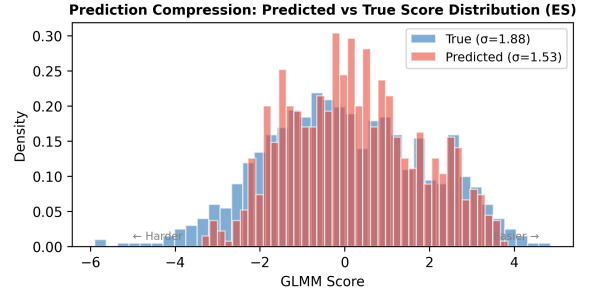


Figure 4: Predicted vs. true GLMM score distributions (ES test set). Predictions are compressed relative to the true distribution.

(ES), -2.00 (DE), and -1.03 (CN), indicating that Spanish speakers find this usage most challenging. Similarly, “received” as an adjective is consistently difficult across all L1s (ES: -5.90 , DE: -5.06 , CN: -3.88), while “clear” as an adverb shows strong L1 variation (ES: -4.14 , DE: $+2.64$, CN: -3.12), suggesting that German speakers find this usage relatively easy compared to other L1 groups.

5.3 Prediction Compression

Our predictions exhibit variance compression: the predicted standard deviation is only 81% of the true value for Spanish ($\sigma_{pred} = 1.53$ vs. $\sigma_{true} = 1.88$; Figure 4). We attempted post-hoc calibration by scaling predictions, but this provided minimal improvement, indicating the gap to top systems lies in prediction quality (Pearson $r = 0.802$ vs. winner’s $r = 0.920$) rather than calibration alone.

5.4 Ridge Blend Failure

One of our open track runs incorporated a Ridge regression with interaction features (frequency $\times$ word length, cognate $\times$ frequency), blended with our ensemble at 85%/15%. While this improved development set performance, it failed on test: RMSE of 1.397 (DE) and 1.421 (CN), worse than the baseline. The interaction features captured patterns in the 677-item development set that did not generalize to the test set. This outcome highlights the risk of incorporating models validated only on a single held-out split without cross-validation. We did not perform cross-validation of the blend weights prior to submission, which we identify as a methodological limitation.

5.5 Gap to Top Systems

Our system explains 64.3% of score variance ($r^2 = 0.643$ for ES) compared to the winners’ 84.6% ($r^2 = 0.846$). We hypothesize that this

gap is driven by three factors: (1) model scale (xlm-roberta-base has 270M parameters, while larger variants such as xlm-roberta-large or DeBERTa-v3-large have 430M–560M); (2) model diversity across multiple architectures; and (3) the use of generative large language models fine-tuned for regression in the open track, in addition to encoder models like ours. These three factors likely account for the bulk of the 0.27–0.40 RMSE gap.

6 Conclusion

We presented a multi-model ensemble for vocabulary difficulty prediction, combining linguistic features with XLM-RoBERTa transformers across both tracks. Our system outperforms baselines on all six track-L1 combinations. Error analysis reveals polysemous words in rare senses as the primary failure mode, with nominalized *-ing* forms showing 58% higher RMSE. The gap to top systems is driven primarily by model scale and diversity. For future work, we identify larger models, multi-architecture stacking, and sense-aware features as the most promising directions.

Limitations

Our system has several limitations. We use only xlm-roberta-base (270M parameters), limiting model capacity. Our multi-seed averaging uses only 3 seeds. Feature engineering operates at the word level without sense-level or morphological features. The Ridge blend was validated on a single split rather than with cross-validation, leading to overfitting. We do not leverage LLMs for the open track. Finally, while we have extended our error analysis to include cross-lingual comparisons, a more systematic analysis across all L1s remains future work.

Ethics Statement

This work uses the publicly available BEA 2026 shared task dataset. The data contains vocabulary items and aggregated difficulty scores; no individual learner data is accessible. Our system is intended to support educational applications. Automated difficulty prediction should complement, not replace, expert judgment in high-stakes assessment.

Acknowledgements

We thank Dr. Shabana K M, Senior Project Scientist at the Wadhvani School of Data Science and AI, Indian Institute of Technology Madras, for her

guidance and feedback on this work. We also thank the shared task organizers for organizing the task and providing the dataset. Computing resources were provided by Kaggle.

References

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the bea 2026 shared task on vocabulary difficulty prediction for english learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. Introducing knowledge-based vocabulary lists (kv1). *Tesol Journal*, 12(4).
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based vocabulary lists*. University of Toronto Press.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.
- Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: A new and improved word frequency database for british english. *Quarterly journal of experimental psychology*, 67(6):1176–1190.

AIDA at BEA 2026 Shared Task 1: A Two-Stage Framework for L1-Aware Vocabulary Difficulty Prediction with Representation Diversity and Residual Calibration

Seok Hyeon Cho, JunHyeok Choi, Sangeun Ji, and Sung Won Han*

AIDA Lab, Department of Industrial Management Engineering, Korea University
{seokces315, urisem, jse1769, swan}@korea.ac.kr

Abstract

We study vocabulary difficulty prediction for second language (L2) learners, a key component for adaptive language learning and assessment. Existing approaches often treat difficulty as an intrinsic property of words or contexts, overlooking representation-dependent variation and learner-specific factors such as L1 transfer.

We participate in the BEA 2026 Shared Task Closed Track using the Spanish (L1) subset of the KVL dataset. We propose a two-stage framework that decouples representation learning from learner-aware calibration. Stage 1 constructs diverse representations using multiple pretrained encoders with varied pooling and prediction strategies, capturing complementary aspects of lexical and contextual complexity. Stage 2 models systematic residual errors with psycholinguistic and cross-lingual features, enabling explicit correction of prediction biases.

Experiments show that our method outperforms strong baselines, improving RMSE ($1.257 \rightarrow 0.976$) and correlation ($0.765 \rightarrow 0.857$). These results highlight the importance of jointly modeling representation diversity and learner-specific effects. Our system ranked 3rd in the official BEA 2026 Shared Task Closed Track.

1 Introduction

Vocabulary is central to language proficiency, shaping what learners can understand and produce. Accurately estimating vocabulary difficulty is essential for level-appropriate content, adaptive assessment, and personalized learning, yet traditional approaches still rely on costly expert judgment and large-scale pretesting.

Recent work leverages pretrained language models and regression methods, but typically treats difficulty as an intrinsic property of words or contexts, overlooking learner-specific variation. In particular, prior approaches do not explicitly

model how difficulty varies with the learner’s first language (L1), despite strong evidence that cross-lingual effects—such as cognates and false friends—systematically influence acquisition.

We revisit vocabulary difficulty prediction from a representation perspective, arguing that difficulty is inherently representation-dependent. Different encoder architectures and aggregation strategies capture complementary aspects of linguistic complexity, making single-model approaches often insufficient in practice.

To address this, we propose a two-stage framework that decouples representation learning from learner-aware calibration via structured residual modeling. Stage 1 constructs representations using multiple pretrained backbones with diverse pooling and prediction heads, aggregating out-of-fold predictions to capture complementary signals. Stage 2 models systematic residuals using psycholinguistic and cross-lingual features, capturing structured biases such as first language (L1)–second language (L2) alignment and exposure differences. This design improves stability and generalization while preserving reliability.

We evaluate our approach on the BEA 2026 Shared Task (KVL dataset), achieving top-tier performance with consistent improvements in RMSE and correlation. Our code and trained models are available online.¹

Our contributions are:

- We show that vocabulary difficulty is inherently representation-dependent, motivating the use of diverse representations.
- We propose a two-stage framework with residual calibration for L1-aware prediction.
- We demonstrate that combining representation diversity with residual calibration improves L1-aware vocabulary difficulty prediction.

*Corresponding author.

¹https://github.com/seokces315/ACL_BEA2026_2SF

2 Related Work

2.1 Lexical Complexity Prediction

Vocabulary difficulty prediction is closely related to lexical complexity prediction (LCP), which extends complex word identification from classification to regression. Early approaches relied on surface features such as frequency and length (Kuperman et al., 2012), while recent work adopts transformer-based contextual representations (Gooding et al., 2020; Shardlow et al., 2021).

Recent studies demonstrate strong performance with pretrained multilingual encoders in cross-lingual settings (Bani Yaseen et al., 2021; Kelious et al., 2024). However, these approaches typically treat complexity as an intrinsic property of words or contexts, largely ignoring learner-specific variation. In particular, they do not explicitly model how difficulty varies with the learner’s linguistic background, which limits their applicability to L2 settings. Our work addresses this limitation by combining learner-aware representations with feature-based calibration for difficulty prediction.

2.2 L1 Transfer and Cross-lingual Effects

L1 transfer plays a central role in second language acquisition, affecting both facilitation and interference (Ellis, 1997). Cognates can ease processing through cross-lingual similarity, whereas false friends introduce systematic errors due to form–meaning mismatches (Van Assche et al., 2009; Schepens et al., 2012).

While cross-lingual effects are widely studied, their integration into lexical difficulty prediction remains limited. Existing approaches often rely on surface similarity (e.g., edit distance), which fails to capture semantic divergence or exposure differences across languages. As a result, they are prone to systematic biases, particularly in cases such as false friends. In contrast, our work goes beyond surface similarity by leveraging cross-lingual and linguistic signals to capture such divergences.

3 Task and Dataset

3.1 Task Definition

The BEA 2026 Shared Task addresses vocabulary difficulty prediction for English learners with diverse L1 backgrounds. The task is formulated as a regression problem, where the goal is to predict a continuous difficulty score for a given English word conditioned on the learner’s L1.

Each instance includes an English word and its associated L1 information (e.g., translation or context), and the target is a psychometrically calibrated difficulty score. We use the Knowledge-based Vocabulary Lists (KVL) dataset (Schmitt et al., 2021, 2024), a multilingual resource annotated with difficulty scores.

3.2 Task Setup

We follow the Closed Track setting, where models are trained only on the provided data for each L1, without using external training data or combining different L1 datasets. Pre-trained Transformer models and standard NLP tools are allowed, while LLMs are prohibited. Public linguistic resources (e.g., WordNet) may be used for feature extraction.

In this work, we focus on the Spanish (L1) subset of the dataset. All models are trained and evaluated exclusively on Spanish learner data, in accordance with the Closed Track constraints.

3.3 Dataset

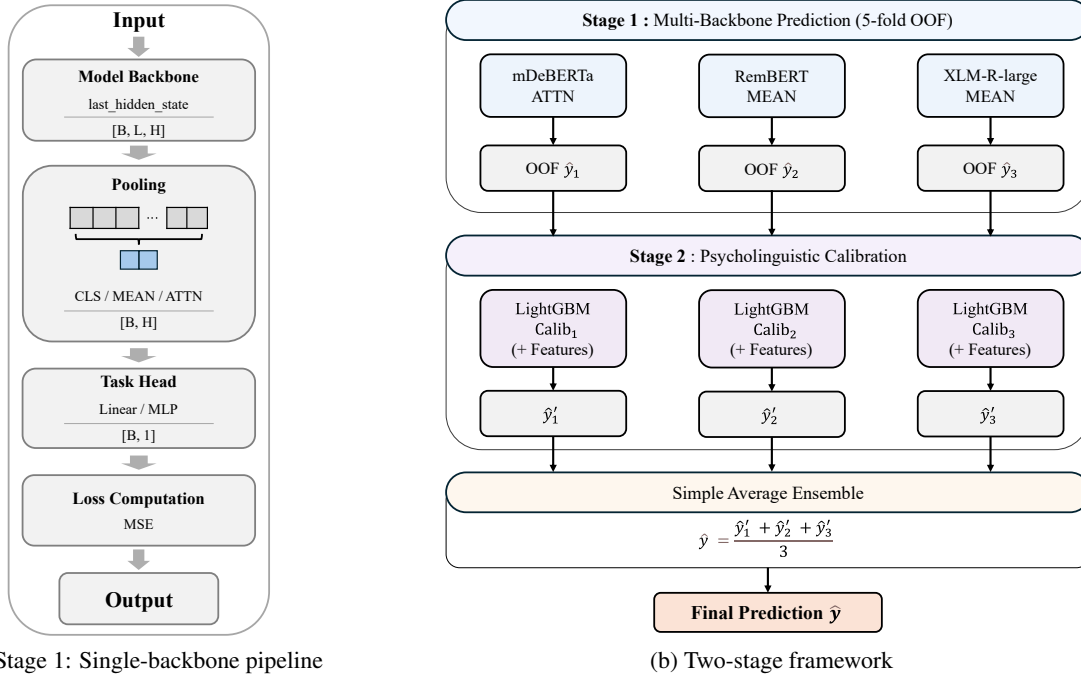
We use an extended version of the British Council’s Knowledge-based Vocabulary Lists (KVL) dataset (Skidmore et al., 2025) to model vocabulary difficulty for English learners with diverse L1 backgrounds. We use only the Spanish subset in our experiments.

The dataset is constructed from large-scale assessments of productive vocabulary knowledge, where learners recall English target words from L1 prompts in a translation-based format (Laufer and Goldstein, 2004). Each item consists of an L1 context sentence, an L1 cue word, and a partially masked English target (e.g., h_____ for “house”).

Difficulty scores are estimated from approximately 3.3 million learner responses using random-item-random-person (RPRI) Rasch models within a generalized linear mixed model (GLMM) framework. The resulting GLMM score (logit scale) is used as the prediction target, where higher values indicate greater difficulty for learners.

The dataset is provided per L1 and split into train/dev/test sets (6,091 / 677 / 748 items). Each instance includes the target word, part-of-speech, a partial spelling clue, the L1 word and context, and a difficulty score. All data are released under CC BY-NC 4.0, with official baselines and evaluation scripts.²

²<https://github.com/britishcouncil/bea2026st>



(a) Stage 1: Single-backbone pipeline

(b) Two-stage framework

Figure 1: Overview of the proposed two-stage framework. (a) Stage 1 uses a single-backbone pipeline (encoder, pooling, and head) with multiple PLMs to produce diverse out-of-fold (OOF) predictions. (b) Stage 2 refines these via residual calibration with psycholinguistic features, followed by simple averaging ensemble.

4 Method

4.1 Stage 1: Multi-Backbone Representations

In the first stage, we construct a diverse set of sequence representation models based on multiple pretrained language model (PLM) backbones and pooling strategies. The goal of this stage is to obtain robust and complementary predictions by leveraging representation diversity across different model architectures and aggregation mechanisms. Since the task requires both lexical cues and contextual understanding, we adopt heterogeneous backbones to capture multi-level signals. This is particularly important for our task, where difficulty prediction requires capturing both surface-level patterns and deeper semantic features that individual models fail to consistently handle.

PLM Backbones. We utilize a diverse set of multilingual encoder-based pretrained language models (PLMs), including XLM-R (Conneau et al., 2020), RemBERT (Chung et al., 2021), mDeBERTa (He et al., 2023), mmBERT (Marone et al., 2025), and mBERT (Devlin et al., 2019), and select a subset based on validation performance and prediction diversity. We observe that their outputs vary for the same input, indicating complementary behavior.

Pooling Strategies. Given an input sequence, each backbone model produces token-level contextual embeddings. Since downstream prediction requires a fixed-length representation, we apply pooling over the final hidden states to construct a sequence-level representation. Specifically, we consider three pooling strategies: the [CLS] token representation, mean pooling, and attention pooling (Lin et al., 2017). We explore various combinations of pooling methods and prediction heads, and select a subset based on validation performance. This design enables the model to capture both global context and salient token-level signals.

Selected Model Variants. We retain three representative models, each with a distinct pooling and head configuration, and use them as base predictors in the subsequent ensemble stage (Figure 1).

Training Protocol. To ensure robust and unbiased predictions, we adopt 5-fold cross-validation (CV) for each model variant. Models are trained on different splits, and out-of-fold (OOF) predictions are generated for all training instances. These OOF predictions ensure each instance is represented by predictions from models not trained on it, preventing data leakage and reducing overfitting bias in Stage 2. The resulting OOF outputs from all base models serve as inputs to the ensemble stage.

4.2 Stage 2: Psycholinguistic Calibration

While Stage 1 captures contextual and semantic representations, it underrepresents linguistically grounded signals (e.g., frequency, morphology, and L1-driven orthographic patterns) related to word difficulty. To address this limitation, Stage 2 leverages handcrafted features and trains a LightGBM regressor (Ke et al., 2017) to predict the residual ($y - \hat{y}_1$), where $\hat{y}_1$ is the Stage 1 prediction. This residual formulation refines predictions without overriding accurate estimates, yielding more stable improvements and lower variance across folds.

Rather than directly predicting the target value, this residual formulation focuses on correcting errors from Stage 1, allowing the model to refine predictions without overriding already accurate estimates. This leads to more stable performance improvements and empirically reduced variance across folds compared to direct regression.

Feature Engineering. We derive features from four linguistically motivated groups, with cross-lingual features capturing L1-positive transfer by approximating how Spanish speakers transform English words orthographically. Specifically, we apply rule-based modifications to the English target, such as *-tion* $\rightarrow$ *-cion* and *ph* $\rightarrow$ *f*, including suffix, phonetic, and orthographic adjustments. We then compute the reduction in Levenshtein distance to the Spanish equivalent before and after transformation, capturing the degree of morphological predictability under L1 influence. We further incorporate a cognate similarity score based on positional character n-gram overlap to capture systematic cross-lingual similarity.

Psycholinguistic features primarily include the log-frequency gap between L2 words and their L1 equivalents and concreteness scores, reflecting cognitive processing difficulty. Lexical and morphological features provide complementary signals, including word frequency, length, and morpheme structure. Full details of cross-lingual transformations are provided in Appendix A.1.

Optimization Strategy. To identify an effective feature configuration, we perform a group ablation study over 119 valid combinations of feature subgroups, evaluated using out-of-fold (OOF) RMSE. We consistently observe that cross-lingual features are included in all competitive configurations, highlighting their central role.

Based on this analysis, we select a subset com-

binning lexical, morphological, cross-lingual, and psycholinguistic features. Although this configuration does not achieve the best performance under default settings, it shows stronger gains after tuning and becomes optimal. This suggests that configurations with psycholinguistic features are more responsive to tuning.

We perform hyperparameter optimization with Optuna, tuning key parameters such as tree complexity and regularization. This achieves the best post-tuning performance across candidate groups.

We also explore an end-to-end alternative that directly integrates handcrafted features into Stage 1, but this approach does not yield consistent or substantial improvements over the standalone neural model. This suggests a mismatch between PLM representations and handcrafted feature distributions, making direct integration less effective. In contrast, the proposed two-stage design separates representation learning from calibration, allowing Stage 2 to model non-linear feature interactions and correct errors more effectively.

Post-processing. We apply a lightweight post-processing pipeline with an OOF-gated selection strategy. Candidate transformations (e.g., clipped blending, isotonic regression, and tail calibration) are sequentially and systematically evaluated and retained only if they strictly reduce OOF RMSE, ensuring consistent gains without overfitting. In practice, only isotonic regression is selected, yielding additional improvements.

Full implementation details, ablation results, and optimization are provided in the Appendix.

5 Experimental Setup

5.1 Data Usage

We use the official train/dev/test splits with model selection on the development set. Each instance is formatted as a natural language prompt combining the L1 context, source word, English target, auxiliary clues, and part-of-speech, as shown in Table 9 in Appendix A.4, and fed into the PLM.

5.2 Evaluation Metrics

We evaluate performance using RMSE (Root Mean Squared Error) between predicted and gold GLMM scores. RMSE is the primary metric for model comparison. We also report the Pearson correlation coefficient to measure the linear relationship between predictions and ground-truth scores.

5.3 Baselines and Ablation Models

We compare our method with external baselines and internal ablation models.

As baselines, we consider (1) the official baseline provided by the shared task organizers and (2) a feature-based Ridge regression model using hand-crafted features, including cross-lingual exposure, contextual average word length, frequency-based statistics (e.g., Zipf score, frequency rank, L1/L2 frequency), and concreteness.

We also include ablation models corresponding to the first stage of our framework. Stage 1 (Single) denotes single-backbone models with different pooling and head configurations, each evaluated independently; performance is reported as the average. Stage 1 (Ensemble) aggregates predictions from the top three encoder backbones, each paired with its best pooling and head configuration.

These variants analyze the contribution of model diversity and the effectiveness of the two-stage design. Final results are presented in Section 6.1.

6 Results

6.1 Main Results

We evaluated the proposed approach against the official baseline and progressively stronger model variants to assess the contribution of each component. Table 1 summarizes the overall performance.

Model	Dev RMSE ↓	Test	
		RMSE ↓	r ↑
<i>Baselines</i>			
Official Baseline	—	1.257	0.765
Feature-based (Ridge)	1.426	1.429	0.652
<i>Ablations</i>			
Stage 1 (Single)	1.040	1.053	0.838
Stage 1 (Ensemble)	0.975	0.989	0.851
Two-Stage (Ours)	0.958	0.976	0.857

Table 1: Performance comparison across baselines and ablation models. RMSE is reported for the development set, and both RMSE and Pearson correlation (r) are reported for the test set.

Our method clearly outperformed all baselines. Compared to the official baseline, the proposed two-stage model significantly reduced RMSE (1.257 $\rightarrow$ 0.976) while improving Pearson correlation (0.765 $\rightarrow$ 0.857), indicating better alignment with ground-truth difficulty scores.

A large performance gap between the baselines (both official and feature-based) and Stage 1 (Sin-

gle) highlighted the importance of task-specific tuning of pretrained encoders with appropriate pooling strategies and prediction heads.

We observed consistent gains across modeling stages. Moving from Stage 1 (Single) to Ensemble improved performance, highlighting representation diversity, while the two-stage model further enhanced accuracy and stability through refinement.

Overall, the results suggested that the primary gains stemmed from effective representation learning and diversity, while additional improvements were achieved through explicit refinement in the second stage. Consistent with these findings, our approach ranked 3rd on the official shared task leaderboard.

6.2 Representation Analysis and Ensembling

To analyze representation design, we evaluated multiple encoder families (BERT, mDeBERTa, RemBERT, and XLM-R) across pooling and prediction head configurations. For each encoder, we report both the average RMSE across all configurations (as a robustness proxy) and the best RMSE.

As shown in Table 2, XLM-R (large) achieved the best performance (1.029 best, 1.038 avg), followed by RemBERT and mDeBERTa-v3.

Encoder	Dev RMSE ↓	
	Avg	Best
XLM-R (large)	1.038	1.029
RemBERT	1.045	1.036
mDeBERTa-v3	1.057	1.047
mmBERT	1.072	1.062
XLM-R (base)	1.193	1.185
mBERT (cased)	1.267	1.254

Table 2: Encoder comparison across pooling and head configurations. We report mean RMSE (Avg; reflecting robustness across configurations) and best RMSE for each encoder, highlighting encoder-dependent performance variation.

Despite the general advantage of larger encoders, representation design introduced substantial variation: within a single encoder, performance differences across configurations reached 0.02–0.04 RMSE, comparable to or exceeding inter-encoder gaps (e.g., 0.007 between XLM-R (large) and RemBERT). This indicated that encoder choice alone was insufficient, and pooling/head design played an equally critical role.

Table 3 presents representative top configurations for each encoder, suggesting encoder-specific

preferences. For instance, XLM-R (large) worked best with mean pooling and a linear head, whereas mDeBERTa-v3 favored attention-based pooling.

Encoder	Pooling + Head (Dev RMSE ↓)		
	MEAN+Lin	ATTN+Lin	ATTN+MLP
XLM-R (large)	1.029	1.031	1.041
RemBERT	1.036	1.048	1.042
mDeBERTa-v3	1.050	1.047	1.063

Table 3: Representative top-performing pooling and head configurations for each encoder. The results indicate encoder-dependent preferences over design choices, highlighting systematic differences in preferred strategies across models.

Moreover, configuration rankings varied across encoders, suggesting that models captured different aspects of the task in practice. A single unified design would therefore underutilize encoder-specific strengths. Instead, we adopted a model-wise optimal strategy, pairing each encoder with its best-performing configuration.

Our final ensemble integrated the top three encoders—XLM-R, RemBERT, and mDeBERTa-v3—each paired with its optimal pooling and head configuration. This design leveraged diverse representations across models, consistently leading to improved overall performance.

Overall, our results highlighted that performance gains stemmed not only from stronger pretrained backbones, but also from effectively leveraging representation diversity inherently induced by pooling and head design choices.

6.3 Calibration and Residual Correction

We constructed the Stage 1 ensemble by averaging three models—XLM-R (large), RemBERT, and mDeBERTa-v3—each with its optimal pooling and prediction head. Although effective, the ensemble still exhibited structured residual errors. To address this, we applied calibration in the prediction space (Guo et al., 2017). Table 4 summarizes the results.

The uncalibrated ensemble achieved 1.028 OOF RMSE and 0.995 Dev RMSE. Calibration consistently improved performance. Under Avg-then-Calib, RMSE decreased to 0.999 (LightGBM), indicating that residual errors remained learnable after ensembling.

Calib-then-Avg yielded larger gains, outperforming Avg-then-Calib across all calibrators. The best performance was achieved by LightGBM, reaching 0.976 OOF RMSE and 0.958 Dev RMSE.

Strategy	Calibrator	RMSE ↓	
		OOF	Dev
No calibration	–	1.028	0.995
Avg-then-Calib	Ridge	1.017	0.984
	XGBoost	1.011	0.974
	LightGBM	0.999	0.977
Calib-then-Avg	Ridge	1.013	0.978
	XGBoost	0.995	0.968
	LightGBM	0.976	0.958

Table 4: Comparison of calibration strategies for the Stage 1 ensemble. Avg-then-Calib denotes averaging followed by calibration, while Calib-then-Avg denotes calibrating before averaging. OOF RMSE is computed on out-of-fold predictions.

These results suggested that calibrating individual models before ensembling was more effective, correcting model-specific residual patterns prior to aggregation. Additionally, tree-based calibrators (LightGBM, XGBoost) outperformed Ridge, highlighting the importance of modeling nonlinear residuals. Based on this, we adopted Calib-then-Avg with LightGBM as the final approach.

6.4 Performance on Tail Distributions

Modeling extreme cases was a key challenge in continuous difficulty estimation. To analyze this, we evaluated performance across difficulty tiers, treating Easy and Hard as tails and Mid as center.

Table 5 reports RMSE for Stage 1 (S1) and Stage 2 (S2), along with relative changes, further stratified by cognate status. Difficulty groups are defined as Easy ($y > 2.5$), Mid ($-2.0 < y \leq 2.5$), and Hard ($y \leq -2.0$), and N denotes the number of samples in each subset. All rows summarize group-level performance.

Difficulty	Cognate	N	Dev RMSE ↓		Δ (%)
			S1	S2	
Easy	All	621	1.019	1.047	-2.75
	Cog.	170	0.985	0.945	+4.06
	Non-Cog.	451	1.031	1.083	-5.04
Mid	All	4537	0.881	0.847	+3.86
	Cog.	626	0.920	0.872	+5.22
	Non-Cog.	3911	0.874	0.843	+3.55
Hard	All	933	1.450	1.369	+5.59
	Cog.	50	1.833	1.794	+2.13
	Non-Cog.	883	1.425	1.341	+5.89

Table 5: Performance across difficulty groups and lexical similarity subsets. S1 and S2 denote Stage 1 and Stage 2 models, respectively. Δ (%) indicates RMSE reduction from S1 to S2 (positive = better).

Results showed a clear asymmetry across regions. In the Mid tier, S2 consistently improved performance (All: +3.86%, Cognates: +5.22%, Non-cognates: +3.55%), indicating effective residual correction in stable regions where feature signals were well-aligned.

In contrast, performance degraded in the Easy tail (All: -2.75%), mainly due to non-cognates (-5.04%), despite cognate gains (+4.06%). This suggested over-correction in low-variance regions where predictions were already reliable, likely due to over-reliance on similarity-based features.

At the Hard tail, S2 yielded the largest improvements (All: +5.59%, Cognates: +2.13%, Non-cognates: +5.89%), indicating that residual modeling effectively reduced errors in high-uncertainty cases, including challenging *false-friend* scenarios where surface similarity was misleading.

Overall, calibration yielded the greatest improvements in high-uncertainty regions, where predictions were less reliable, but might lead to overcorrection in low-variance regions, where predictions were already stable, thereby underscoring the need for difficulty-aware calibration.

7 Discussion

7.1 Two-Stage Modeling

A central finding was that the proposed framework constituted a structural decomposition between representation and calibration. Rather than attempting to jointly model all factors in a single space, the two-stage design separated robust difficulty representation from systematic error correction.

A key insight underlying Stage 1 was that difficulty representation was fundamentally model-dependent. Different encoder architectures and aggregation choices captured complementary aspects of linguistic complexity, indicating that no single configuration was universally optimal. As a result, relying on a fixed architecture risked overlooking important signals. Instead, Stage 1 constructed a robust representation layer by integrating strong configurations, enabling the model to capture diverse facets of language-intrinsic difficulty while reducing sensitivity to individual design choices.

This design was motivated by the observation that optimal configurations varied across encoders, highlighting the importance of encoder-specific optimization for stable representation learning.

Stage 2 operated in the residual space, performing learner-dependent and psycholinguistic correc-

tion. By incorporating language-specific signals (e.g., cognate effects and cross-lingual similarity patterns) grounded in the Spanish domain, it addressed systematic discrepancies—such as mismatches between surface similarity and actual difficulty—that were not captured by semantic representations alone.

Crucially, calibration in this framework functioned as a structured bias correction mechanism. By decoupling representation from correction, the model preserved the underlying difficulty signal while explicitly modeling residual errors, which led to more stable and interpretable predictions, particularly in challenging cases where single-stage models tended to conflate linguistic similarity with true difficulty.

7.2 Feature Importance Analysis

The tree-based calibration model enabled interpretability via feature importance measured by split frequency. To support reproducibility and clarify the feature design, a complete inventory of handcrafted features and their sources is provided in Table 11.

As shown in Table 6, top features fell into three categories—cross-lingual, lexical, and psycholinguistic. We utilized handcrafted features based on lexical frequency information from Wordfreq (Speer, 2022), psycholinguistic properties such as concreteness (Brysbaert et al., 2014), word complexity, and cross-lingual exposure derived from English–Spanish frequency differences.

Feature	Category	Split Count
Crossling_Exposure	Cross-lingual	180
Ctx_Avg_Word_Len	Psycholinguistic	176
Global_Zipf_Score	Lexical	171
Freq_Rank_Norm	Lexical	160
Freq_L1_L2_Gap	Psycholinguistic	157
L1_Zipf_Score	Psycholinguistic	143
Concreteness	Psycholinguistic	136

Table 6: Top features ranked by importance based on split frequency in the tree-based model. Feature names are abbreviated; detailed descriptions are provided in the text. *Split Count* denotes the number of times a feature is selected for node partitioning across all trees.

Psycholinguistic and exposure-related features dominated: four of the top seven were psycholinguistic, with others reflecting frequency or cross-lingual exposure. Importantly, *Crossling_Exposure* ranked highest, followed by *Global_Zipf_Score* and *Freq_L1_L2_Gap*.

These results suggested that difficulty for Spanish L1 learners was driven less by intrinsic word properties and more by exposure and alignment with prior linguistic experience. The importance of *Freq_L1_L2_Gap* highlighted cross-lingual frequency mismatch. Finally, *Concreteness* supported prior findings that concrete words were easier to acquire (Brysbaert et al., 2014; Paivio, 1991).

7.3 Error Analysis

To analyze model limitations, we examined representative failure cases from two complementary perspectives: systematic errors due to false friends and qualitative examples of model behavior.

A primary source of error arose from *false friends*, where words were orthographically similar to their Spanish counterparts but semantically divergent. In such cases, the model overestimated easiness due to surface-form similarity (e.g., *embarrassed* vs. *embarazada*). This suggested that while cross-lingual similarity was effectively captured, semantic alignment was insufficiently modeled, particularly in semantically divergent cases.

Table 7 shows representative examples of Stage 2 calibration. GLMM denotes ground-truth difficulty, and S1/S2 indicate predictions from Stage 1 and Stage 2, respectively. *Cognates* (↑) were consistently adjusted toward easier difficulty, *Non-cognate* instances remained appropriately difficult without substantial change, and *Failure* highlighted errors arising from missed cognate detection.

English	Spanish	GLMM	S1	S2
<i>Cognates</i> (↑)				
domination	dominación	+1.40	-0.36	+0.75
consecutive	consecutivo	+1.19	-0.62	+0.44
discipline	disciplina	+2.06	+0.51	+1.56
compulsive	compulsivo	+1.79	-0.25	+0.75
selective	selectivo	+2.13	-0.07	+0.92
<i>Non-cognate</i>				
skillet	sartén	-3.02	-1.08	-2.56
<i>Failure</i>				
synonym	sinónimo	-1.72	+0.83	-0.26

Table 7: Stage 2 calibration examples by word type. GLMM: ground truth; S1/S2: Stage 1/2 predictions.

For transparent cognates (e.g., *domination* / *dominación*), Stage 1 underestimated easiness, whereas Stage 2 corrected predictions using additional features. Non-cognates (e.g., *skillet*) remained appropriately difficult, indicating that calibration was not applied indiscriminately.

However, failure cases (e.g., *synonym* / *sinónimo*) revealed limitations of the cognate detection mechanism. Despite semantic equivalence, orthographic mismatch led to low similarity scores and inaccurate predictions.

Overall, while the model effectively leveraged cross-lingual similarity, it remained vulnerable to semantic divergence and surface-form limitations.

8 Conclusion

In this work, we addressed the problem of vocabulary difficulty prediction for L2 learners by focusing on representation diversity and learner-specific factors. We proposed a two-stage framework that separates representation learning from psycholinguistic calibration, enabling the model to capture both language-intrinsic difficulty and systematic cross-lingual effects.

Experimental results demonstrated that our approach consistently outperformed strong baselines, achieving substantial improvements in both RMSE and correlation. The primary gains stemmed from combining diverse pretrained encoders with optimized pooling and prediction head configurations, allowing the model to capture complementary lexical, contextual, and semantic signals. The calibration stage further improved predictions by modeling structured residual errors using psycholinguistic and cross-lingual features, particularly in high-uncertainty regions where semantic representations alone were insufficient.

Our analysis further suggested that vocabulary difficulty should not be viewed solely as an intrinsic property of words, but as a learner-dependent phenomenon shaped by prior linguistic experience and L1-L2 alignment. More specifically, features related to cross-lingual exposure and frequency mismatch played a central role in explaining learner difficulty for Spanish L1 speakers.

Overall, the proposed framework demonstrates that separating representation learning from residual correction provides a stable and effective approach for L1-aware vocabulary difficulty prediction. Beyond the shared task setting, we believe this framework offers a practical foundation for adaptive language learning and personalized educational applications.

Future work will explore semantics-aware cross-lingual representations and more adaptive calibration strategies to better handle semantically challenging cases across diverse learner settings.

9 Limitations

Despite the strong overall performance, our framework still exhibited several limitations. First, the model remained vulnerable to cases involving semantic divergence, such as false friends, where reliance on surface-level similarity occasionally produced systematic prediction errors. This suggests that the current representation space does not fully capture deeper semantic correspondence across languages. In addition, the calibration stage sometimes introduced over-correction in low-variance regions, indicating that more adaptive calibration strategies may be necessary for stable refinement.

We also note that the exploration of Stage 1 ensemble configurations was not fully exhaustive due to computational limitations and time constraints. Finally, although the proposed framework demonstrated strong effectiveness for Spanish L1 learners, its generalizability to broader learner populations and additional L1 settings remains to be validated.

References

- Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Es-lam 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.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. [Concreteness ratings for 40 thousand generally known english word lemmas](#). *Behavior Research Methods*, 46(3):904–911.
- Joanne F. Carlisle. 2000. [Awareness of the structure and meaning of morphologically complex words: Impact on reading](#). *Reading and Writing*, 12(3–4):169–190.
- Hyung Won Chung, Thibault Févry Lee, Yingjie Huang, Andrew Dai, and Saliou Adafre. 2021. Rethinking embedding coupling in pre-trained language models. In *Proceedings of the International Conference on Learning Representations*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- David Crystal. 2003. *The Cambridge Encyclopedia of the English Language*, 2 edition. Cambridge University Press, Cambridge.
- 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*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Rod Ellis. 1997. *Second Language Acquisition*. Oxford University Press, Oxford.
- Sian Gooding, Shiva Taslimipoor, and Ekaterina Kochmar. 2020. [Incorporating multiword expressions in phrase complexity estimation](#). In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REading Difficulties (READI)*, pages 14–19, Marseille, France. European Language Resources Association.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the International Conference on Machine Learning*. PMLR.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *Proceedings of the International Conference on Learning Representations*.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, volume 30.
- Abdelhak Keliou, Mathieu Constant, and Christophe Coeur. 2024. [Investigating strategies for lexical complexity prediction in a multilingual setting using generative language models and supervised approaches](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 96–114, Rennes, France. LiU Electronic Press.
- Michael J. Kieffer and Nonie K. Lesaux. 2012. [Development of morphological awareness and vocabulary knowledge in spanish-speaking language minority learners: A parallel process latent growth curve model](#). *Applied Psycholinguistics*, 33(1):23–54.
- Donald E. Knuth. 1973. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, Reading, Massachusetts.
- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. [Age-of-acquisition ratings for 30,000 english words](#). *Behavior Research Methods*, 44(4):978–990.
- Batia Laufer and Zahava Goldstein. 2004. Testing vocabulary knowledge: Size, strength, and computer adaptiveness. *Language Learning*, 54(3):399–436.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Zhouhan Lin, Minwei Feng, Cicero Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. [A structured self-attentive sentence embedding](#). In *Proceedings of the International Conference on Learning Representations*.

M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, and Benjamin Van Durme. 2025. [mm-BERT: A modern multilingual encoder with annealed language learning](#). *arXiv preprint arXiv:2509.06888*.

George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Communications of the ACM*, 38(11):39–41.

Allan Paivio. 1991. [Dual coding theory: Retrospect and current status](#). *Canadian Journal of Psychology*, 45(3):255–287.

Job Schepens, Ton Dijkstra, and Frank Grootjen. 2012. [Distributions of cognates in europe as based on levenshtein distance](#). *Bilingualism: Language and Cognition*, 15(1):157–166.

Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. [Introducing knowledge-based vocabulary lists \(kvl\)](#). *TESOL Journal*, 12(4):e622.

Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based Vocabulary Lists*. University of Toronto Press.

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.

Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.

Robyn Speer. 2022. [wordfreq: Frequency data for natural language processing](#).

Eva Van Assche, Wouter Duyck, and Robert J. Hart-suiker. 2009. [The cognitive mechanisms of bilingual reading: The case of cognate facilitation](#). *Journal of Memory and Language*, 60(1):92–107.

William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the Fellegi–Sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods of the American Statistical Association*, pages 354–359.

A Appendix

A.1 Cross-lingual Feature Details

A.1.1 Hispanized Transformation Rules

To approximate the orthographic transformation process of Spanish learners, we apply a sequence of rule-based modifications to English target words. The rules are grouped as follows:

Suffix Mapping. We map common English suffixes to their Spanish counterparts:

- *-tion* → *-cion*
- *-ty* → *-dad*
- *-ly* → *-mente*
- *-ous* → *-oso*
- *-ence* / *-ance* → *-encia* / *-ancia*

Phonetic Mapping. We simulate phonological adaptation:

- *ph* → *f*
- *th* → *z* (Peninsular) or *s* (Latin American seseo)

Prothetic *e*. We prepend *e* to words beginning with an *s*+consonant cluster:

- *special* → *especial*

Consonant Simplification. We reduce double consonants except *ll* and *rr*.

A.1.2 Distance-Based Feature

Let d_{orig} denote the Levenshtein distance between the original English word and its Spanish translation, and d_{trans} denote the distance after applying the transformation rules. We define the cross-lingual predictability feature as:

$$\Delta d = d_{\text{orig}} - d_{\text{trans}} \quad (1)$$

A larger Δd indicates that the transformed form is closer to the Spanish equivalent, reflecting higher morphological predictability under L1 influence.

A.1.3 Cognate Similarity

We compute a positional character n-gram overlap score to capture cognate similarity. The score is defined as a weighted sum of matching character sequences:

- First-character match: +0.3

- First three-character match: +0.4
- Last three-character match: +0.3

Words with a score above 0.4 are considered cognates. We additionally include a binary feature indicating systematic cognate correspondence.

A.2 Feature Group Ablation

To analyze the contribution of each feature group, we perform a group ablation study over combinations of feature subgroups and evaluate them using out-of-fold (OOF) RMSE. Table 8 reports the top-performing configurations.

Feature Groups	OOF RMSE ↓
lex + cross + derived	1.0506
lex + morph + cross + derived	1.0508
lex + morph + cross	1.0510
lex + cross	1.0511
lex + morph + cross + psycho	1.0513

Table 8: Top-5 feature group combinations from the ablation study. Cross-lingual features consistently appear in all competitive configurations. The selected setting (lex + morph + cross + psycho) is used for Stage 2.

We consistently observe that cross-lingual features are present across all high-performing configurations, confirming their central role in modeling word difficulty.

A.3 Optimization and Postprocessing Details

A.3.1 Hyperparameter Optimization

We conduct hyperparameter optimization using Optuna to tune a set of key LightGBM parameters such as `n_estimators`, `learning_rate`, `num_leaves`, `min_child_samples`, `subsample`, `colsample_bytree`, and L1/L2 regularization.

The selected configuration achieves the best post-tuning performance across all candidate feature groups, with an OOF RMSE of 1.040.

A.3.2 Post-processing Details

We apply a sequence of candidate post-processing steps to further calibrate model predictions. Each step is evaluated using out-of-fold (OOF) RMSE and is only retained if it strictly improves performance.

Candidate Steps. The post-processing pipeline includes clipped blending, isotonic regression, and tail calibration.

Selection Outcome In practice, only isotonic regression satisfies the OOF gating criterion, resulting in an additional reduction of approximately 0.010 in OOF RMSE.

A.4 Prompt Construction

We construct the input as a structured natural language prompt by mapping dataset fields to semantically meaningful components. The *Spanish context* provides the L1 sentence containing the target word, while the *Spanish word* denotes the aligned lexical item. The *English translation* specifies the target whose difficulty is to be predicted. The *Meaning clue* provides auxiliary hints, with part-of-speech information appended in parentheses. The prediction target is the GLMM score, which reflects the perceived difficulty of the English word for L1 speakers. Table 9 shows the mapping from each component to its textual realization.

Component	Example
Spanish context	El eclipse solar fue visible durante un breve lapso de tiempo.
Spanish word	lapso
English translation	span
Meaning clue	s___ (noun)

Table 9: Example of the structured prompt, showing the mapping from each component to its textual realization.

B Training Hyperparameters

Setting	Value
<i>Shared settings</i>	
Optimizer	AdamW
Learning rate	2×10^{-5}
Batch size	16
Gradient accumulation steps	2
Warmup ratio	0.1
Weight decay	0.01
Max sequence length	256
Precision	FP16
<i>LightGBM calibrator (Optuna-tuned)</i>	
<code>n_estimators</code>	400
<code>learning_rate</code>	0.03
<code>num_leaves</code>	31
<code>min_child_samples</code>	20
<code>subsample</code>	0.8
<code>colsample_bytree</code>	0.8
<code>reg_alpha</code> (L1)	0.1
<code>reg_lambda</code> (L2)	1.0

Table 10: Training hyperparameters for all components of the proposed pipeline.

C Feature Inventory for Stage 2 Calibration

To support reproducibility and clarify the feature design, Table 11 provides a complete inventory of handcrafted features used in the Stage 2 calibration model, including feature descriptions, data sources, and references for externally derived resources.

Feature	Description	Source
<i>Lexical Features (9)</i>		
Global_Zipf_Score	English Zipf frequency	Speer (2022)
Is_Low_Zipf	Low-Zipf flag (< 3.5)	Speer (2022)
Freq_Rank_Norm	Rank-normalised dataset frequency	Dataset
Dataset_Freq_Raw	Raw target frequency	Dataset
Char_Count	Character count	Dataset
Vowel_Ratio	Vowel-to-character ratio	Orthographic
Consonant_Cluster_Max	Maximum consonant cluster length	Orthographic
Has_Double_Letter	Repeated-letter flag	Orthographic
Letter_Rarity	Proportion of rare letters	Orthographic
<i>Morphological Features (8)</i>		
Syllable_Count	Estimated syllable count	Crystal (2003)
Morpheme_Count_Est	Estimated morpheme count	Carlisle (2000)
Suffix_Count	Number of suffix matches	Carlisle (2000)
Prefix_Count	Number of prefix matches	Kieffer and Lesaux (2012)
Morph_Complexity	Prefix/suffix complexity flag	This work
Is_Abstract	Abstract suffix flag	This work
POS_Encoded	Encoded POS category	Dataset
POS_Frequency	POS frequency in training data	Dataset
<i>Cross-lingual Features (18)</i>		
Norm_Lev_Dist	Normalised Levenshtein distance	Levenshtein (1966)
Lev_Dist_Raw	Raw Levenshtein distance	Levenshtein (1966)
Hispanized_Lev_Dist	Hispanized edit distance	This work
Hispanized_Lev_Gain	Hispanization distance reduction	This work
JaroWinkler_Similarity	Jaro–Winkler similarity	Winkler (1990)
Bigram_Overlap	Bigram overlap ratio	String algorithms
Phonetic_Similarity	Soundex-based similarity	Knuth (1973)
LCS_Ratio	Normalised LCS score	Knuth (1973)
LCS_Length	Longest common substring length	Knuth (1973)
Cognate_Score	Heuristic cognate similarity	This work
Systematic_Cognate	EN–ES suffix correspondence	This work
L1_Char_Count	L1 character count	Dataset
Char_Count_Diff	EN–L1 character difference	Dataset
Char_Count_Ratio	EN–L1 character ratio	Dataset
Internal_Polysemy	Distinct L1 cues per EN target	Dataset
L1_Polysemy	Distinct EN targets per L1 cue	Dataset
WordNet_Sense_Count	Number of WordNet synsets	Miller (1995)
Crossling_Exposure	EN Zipf frequency in Spanish corpora	Speer (2022)
<i>Psycholinguistic Features (11)</i>		
Concreteness	Mean concreteness score	Brysbaert et al. (2014)
Is_Highly_Concrete	High-concreteness flag	Brysbaert et al. (2014)
Is_Highly_Abstract	High-abstractness flag	Brysbaert et al. (2014)
WordNet_Max_Depth	Maximum WordNet depth	Miller (1995)
WordNet_Avg_Path_Len	Average WordNet path length	Miller (1995)
L1_Zipf_Score	L1 Zipf frequency	Speer (2022)
Freq_L1_L2_Gap	EN–ES Zipf frequency gap	Speer (2022)
Log_Freq_Band	Discretised Zipf frequency band	Speer (2022)
Ctx_Avg_Word_Len	Average context word length	Dataset
Ctx_TTR	Context type-token ratio	Dataset
Clue_Coverage	Revealed-letter ratio in clue	Dataset
<i>Residual Interaction Feature</i>		
BERT_x_Lev_Dist	Stage 1 prediction $\times$ edit distance	This work

Table 11: Inventory of the 47 handcrafted features used for Stage 2 calibration. “Dataset” denotes features computed directly from the shared-task training data, “Orthographic” refers to deterministic character-level features, and “This work” indicates heuristic features introduced in this work.

Failure at BEA 2026 Shared Task 1: One Pipeline, Three L1s: A Unified Language-Agnostic System for Vocabulary Difficulty Prediction

Abid Al Hossain

Chittagong University of
Engineering and Technology
Chattogram, Bangladesh
aabid.al.hossain@gmail.com

Kamruzzaman Khan Alve

Rajshahi University of
Engineering and Technology
Rajshahi, Bangladesh
kamruzzamanalve@gmail.com

Abstract

We present team Failure’s submission to the BEA 2026 Shared Task on L1-aware vocabulary difficulty prediction for English learners (Felice and Skidmore, 2026). We investigate whether a single, language-agnostic pipeline can achieve competitive performance across typologically distinct L1 settings without per-language manual engineering. We implement a unified notebook, model architecture, and training procedure for Spanish, German, and Mandarin Chinese. Our system fine-tunes xlm-roberta-base (Conneau et al., 2020) with attention-mask-weighted mean pooling, augmented with four universal length-based features. We use Optuna-driven hyperparameter search (Akiba et al., 2019) and a leakage-free 5-fold ensemble with fold-specific standardization. Under this unified and compute-constrained setup, our closed-track system outperforms the official baseline on all three evaluation languages: RMSE 1.219 vs. 1.257 for Spanish, 1.118 vs. 1.258 for German, and 1.090 vs. 1.140 for Chinese. These results suggest that a compact, reproducible pipeline can yield consistent improvements without L1-specific customization, providing a practical baseline for multilingual educational NLP.

1 Introduction

Vocabulary difficulty prediction is a canonical educational NLP task, with applications in adaptive assessment, content selection, and learning models. The BEA 2026 Shared Task (Felice and Skidmore, 2026) poses the task as a regression problem: given an English vocabulary test item and its corresponding first-language (L1) context, the objective is to predict the psychometrically calibrated GLMM difficulty score for learners from one of three L1 backgrounds: Spanish, German, and Mandarin Chinese.

One might imagine solving the task in the most intuitive way, by training specialized models for

each L1 condition, which requires different feature engineering or heuristics tailored to the language at hand. However, we adopt a language-agnostic stance. The motivation for this work is grounded in the assumption that there is sufficient underlying structure in the task across L1 conditions to allow a generalized approach to perform better than the official baseline on all three languages, even in a setup where all models are independently tuned with identical architecture, features, hyperparameters, and training methodology.

There is more to this line of reasoning than maximizing leaderboard scores. A language-agnostic pipeline is more easily audited and reproduced than a language-specific one. It also is more easily extended to new L1 conditions should the community choose to do so in future tasks. A model that can run on freely available hardware is more widely applicable and usable than one requiring institutional compute. Finally, an approach that relies on consistent modeling assumptions rather than language-specific heuristics is a more scientifically interesting contribution.

In this closed-track paper, we show that this language-agnostic pipeline achieves superior performance on all three languages, as predicted. This contribution goes beyond just improving the score, as it provides a complete pipeline that is reproducible in practice and runnable on free GPU hardware.

2 Task and Data

The BEA 2026 Shared Task (Felice and Skidmore, 2026) provides parallel data for three learner L1 conditions: Spanish (es), German (de), and Mandarin Chinese (cn). The target variable is GLMM_score, a psychometric difficulty estimate derived from more than 100,000 learner responses using random-item–random-person Rasch models (Schmitt et al., 2024).

The dataset is based on the British Council’s Knowledge-based Vocabulary Lists (KVL) (Schmitt et al., 2021) and was adapted into the Extended KVL Dataset for NLP (Skidmore et al., 2025). Each item includes the fields `item_id`, `L1`, `en_target_word`, `en_target_pos`, `en_target_clue`, `L1_source_word`, and `L1_context`; labeled splits additionally include `GLMM_score` as the target variable. Items with the same `item_id` across different L1 files are parallel, meaning they correspond to the same English target word but include L1-specific prompts and difficulty annotations.

Each L1 condition contains 6,091 training items, 677 development items, and 748 test items. We train one model per L1 condition using only the corresponding training data. The closed track does not permit combining data across L1s, using external training data, or relying on large language models. Our system fully complies with these constraints.

3 System Overview

Our system is implemented as a single Kaggle notebook and applied to all three L1 conditions. The only run-specific setting is the target L1 identifier (es, de, or cn); no language-specific preprocessing, feature engineering, or post-processing is used.

The pipeline has three stages controlled by a single `MODE` flag: hyperparameter search, 5-fold training with development evaluation, and test inference from saved fold checkpoints. The same codebase is reused across all languages and stages.

Figure 1 summarizes the overall workflow and the shared per-fold model architecture. In `MODE='optuna_tune'`, Optuna searches over the main training hyperparameters using reduced cross-validation. In `MODE='train_dev'`, the selected hyperparameters are used for full 5-fold training with fold-specific standardization, and the best checkpoint is saved for each fold. In `MODE='test_infer'`, the saved checkpoints are loaded, predictions are generated independently per fold, and the final output is obtained by averaging fold predictions and inverse-transforming them to the original `GLMM_score` scale.

4 Input Representation

Text serialization. For each item, we serialize all available text fields into a single input string using bracket markers:

```
[L1]   es   [WORD]  universal  [POS]
adjective [CLUE]  u_____ [SOURCE]
global [CONTEXT] La escalada. . .
```

This jointly exposes lexical, syntactic, orthographic, translational, and contextual information to the encoder. The serialization format is identical for all three languages. Text fields are cleaned by collapsing whitespace and replacing null values with empty strings, ensuring the tokenizer never receives null input.

Numerical features. We extract four character-count features from each item: **word_len**, **clue_len**, **source_len**, and **context_len**, computed as the character length of `en_target_word`, `en_target_clue`, `L1_source_word`, and `L1_context` respectively. These are universally available across all L1s and require no language-specific processing. Length-based signals are present in the task format itself, particularly through the partial-spelling clue, and can provide lightweight complementary information to the encoder.

5 Model Architecture

We use `xlm-roberta-base` (Conneau et al., 2020) as the encoder backbone. The serialized input is tokenized with a maximum length of 128 tokens and passed through the 12-layer transformer. We apply attention-mask-weighted mean pooling over the last hidden states:

$$\mathbf{h} = \frac{\sum_i m_i \cdot \mathbf{e}_i}{\sum_i m_i} \in \mathbb{R}^{768}$$

where $\mathbf{e}_i$ is the i -th token representation and $m_i \in \{0, 1\}$ is the attention mask value, so that padding tokens do not contribute to the pooled representation.

A dropout layer ($p = 0.1$) is applied to $\mathbf{h}$. The four numerical features are independently Z-score standardized using fold-specific training statistics to form $\mathbf{f} \in \mathbb{R}^4$. We concatenate them:

$$\mathbf{z} = [\mathbf{h}; \mathbf{f}] \in \mathbb{R}^{772}$$

A linear regression head produces the predicted difficulty: $\hat{y} = \mathbf{W}\mathbf{z} + b$. Training is performed in standardized target space using mean squared error (MSE) loss, and predictions are inverse-transformed to the original `GLMM` scale at evaluation time:

$$\hat{y}_{\text{raw}} = \hat{y} \cdot \sigma_{\text{fold}} + \mu_{\text{fold}}$$

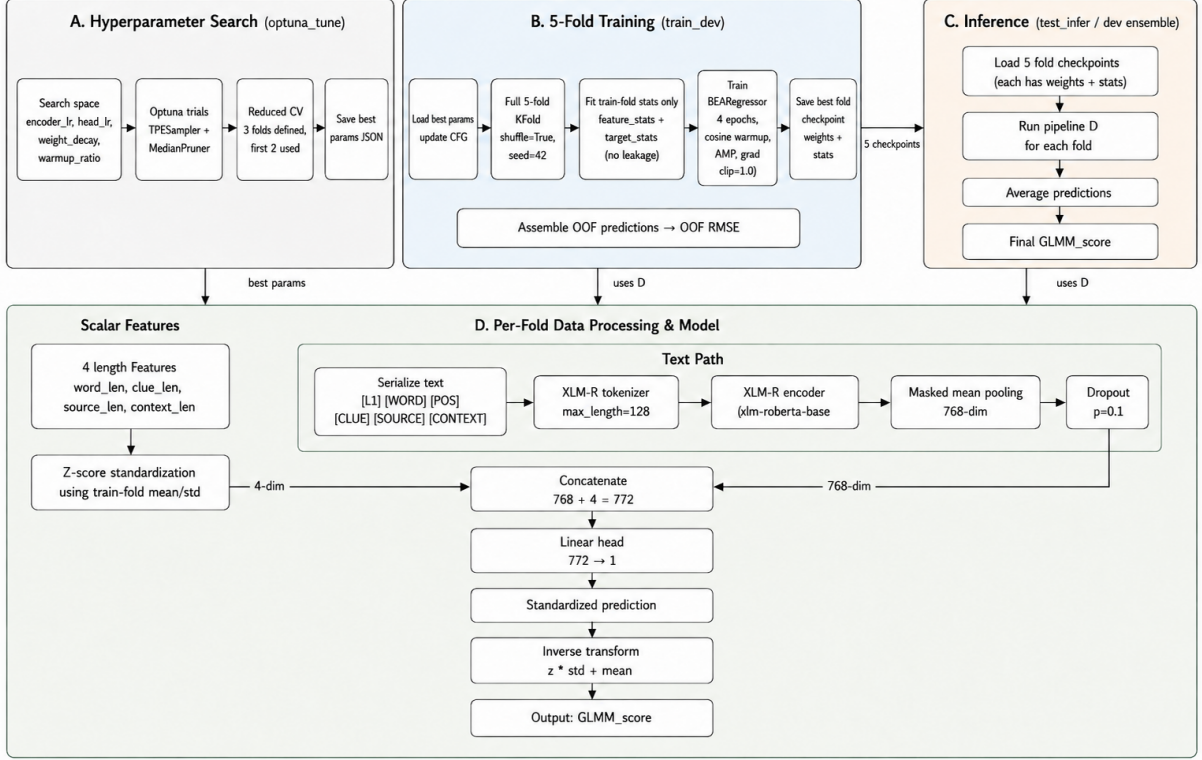


Figure 1: Overview of our unified pipeline for one target L1. The upper panels show the three operating modes, and the lower panel shows the shared per-fold model architecture.

where μ_{fold} and σ_{fold} are the target mean and standard deviation computed on that fold’s training partition.

We selected xlm-roberta-base as the largest multilingual encoder that fit within the memory budget of our two available NVIDIA T4 GPUs.

6 Training Procedure

Hardware and runtime. All experiments were conducted on two NVIDIA T4 GPUs (16 GB each) available through the Kaggle free compute tier. Multi-GPU training was handled automatically via `nn.DataParallel`. A complete run for one L1 condition—including Optuna search, 5-fold training, development evaluation, and test inference—required approximately 2 hours of wall-clock time, with some variation depending on GPU availability.

Hyperparameter optimization. We use Optuna (Akiba et al., 2019) with a TPE sampler (TPESampler, seed 42) and a Median pruner (MedianPruner, `n_startup_trials=5`, `n_warmup_steps=1`) to optimize four hyperparameters: encoder learning rate in $[5 \times 10^{-6}, 5 \times 10^{-5}]$ (log scale), head learning rate in $[3 \times 10^{-5}, 5 \times 10^{-4}]$ (log scale), weight

decay in $[10^{-5}, 10^{-1}]$ (log scale), and warmup ratio in $[0.01, 0.20]$. Each of 15 trials is evaluated using reduced 3-fold cross-validation (KFold, `shuffle=True`, `seed=42`), but only the first 2 folds are scored per trial for efficiency (OPTUNA_N_SPLITS=3, OPTUNA_MAX_FOLDS=2). The objective is the mean validation RMSE across the evaluated folds. Best parameters are saved to a JSON file and automatically loaded before full training, eliminating manual transcription errors.

Leakage-free 5-fold training. For full training, we use 5-fold cross-validation (KFold, `n_splits=5`, `shuffle=True`, `random_state=42`). For each fold, both feature standardization statistics and target standardization statistics are computed only on the fold’s training partition and stored inside the saved checkpoint. These statistics are reused at inference time, preventing any information flow from held-out data. A zero-variance guard ($\sigma < 10^{-8}$) is applied to avoid division errors in degenerate cases.

Optimization. The model is trained with MSE loss in standardized target space. We use AdamW with three parameter groups: encoder param-

ters with the tuned weight decay; LayerNorm and bias parameters without weight decay; and the regression head with its own tuned learning rate and no weight decay. We apply a cosine learning rate schedule with linear warmup (`get_cosine_schedule_with_warmup`), train for 4 epochs, and clip gradients to a maximum norm of 1.0. Mixed precision training (AMP) is enabled throughout via `torch.cuda.amp.autocast` and `GradScaler`. Seeds are fixed (`seed=42`) for approximate reproducibility; cuDNN benchmark mode is enabled for throughput, so results may vary slightly across hardware.

Memory-efficient implementation. To operate within the T4 memory budget, we use dynamic per-batch padding so sequences are padded only to the longest item in each batch. Additional choices include `pin_memory=True` in all `DataLoaders`, `set_to_none=True` in gradient zeroing, explicit GPU cache clearing and Python garbage collection between folds and Optuna trials, and `gc_after_trial=True` in the Optuna study to prevent memory accumulation across trials.

Checkpoint selection and ensemble inference. Within each fold, the checkpoint achieving the lowest validation RMSE across all epochs is saved, rather than the final-epoch state. For each L1, all 5 fold checkpoints are loaded sequentially and their predictions are averaged. Because each checkpoint stores its own feature and target normalization statistics, inference is fully self-contained: no external normalization state is needed beyond what is saved in the checkpoint file.

Fixed settings. Training batch size 32, validation batch size 64, maximum sequence length 128, dropout 0.1, seed 42. All non-tuned settings are identical across Spanish, German, and Mandarin Chinese.

7 Results and Discussion

7.1 Main Results

Table 1 presents our official closed-track test results alongside the official baseline.

Our system improves over the official baseline in RMSE for all three L1 conditions. The same pipeline is applied across all three languages without any language-specific adaptation. The largest absolute improvement is observed for German (0.140 RMSE, 11.1%), followed by Chinese (0.050,

L1	Ours	Baseline	Δ RMSE	Δ %
Spanish	1.219	1.257	-0.038	-3.0%
German	1.118	1.258	-0.140	-11.1%
Chinese	1.090	1.140	-0.050	-4.4%

Table 1: Official closed-track RMSE on the test set. Lower is better.

L1	F1	F2	F3	F4	F5	Mean	Std
Spanish	1.270	1.231	1.269	1.335	1.279	1.277	0.036
German	1.161	1.189	1.168	1.198	1.231	1.189	0.025
Chinese	1.105	1.134	1.106	1.139	1.109	1.118	0.015

Table 2: Per-fold validation RMSE across 5 folds for each L1 condition.

4.4%) and Spanish (0.038, 3.0%). These results show that a shared architecture and training recipe can produce consistent gains across all three L1 settings without any language-specific engineering.

7.2 Validation Stability

Table 2 reports per-fold validation RMSE across all five folds for each L1 condition. Standard deviation remains low for all three languages (0.015–0.036), indicating stable training and suggesting that no single fold dominates the final ensemble prediction. Chinese is the most stable condition, while Spanish shows the largest fold-to-fold variation. Fold 4 yields the highest validation RMSE for both Spanish and German, but the overall spread remains limited, supporting the robustness of the cross-validation procedure.

7.3 Error Analysis

Figure 2 shows out-of-fold mean absolute error (MAE) by gold difficulty band for all three L1 conditions.

Across all three languages, error is highest in the easiest band (≤ -2), at approximately 1.57 for Spanish, 1.59 for German, and 1.52 for Chinese. In the middle bands (-1 to $+1$), MAE is substantially lower. Error increases again at the hardest end (> 2), especially for Spanish.

The same pattern appears across all three L1 conditions, suggesting that the model is less accurate at the extremes of the difficulty range and tends to pull very easy items toward the center of the score distribution.

7.4 Discussion

Language-agnostic generalization. Our system improves on the official baseline for all three L1

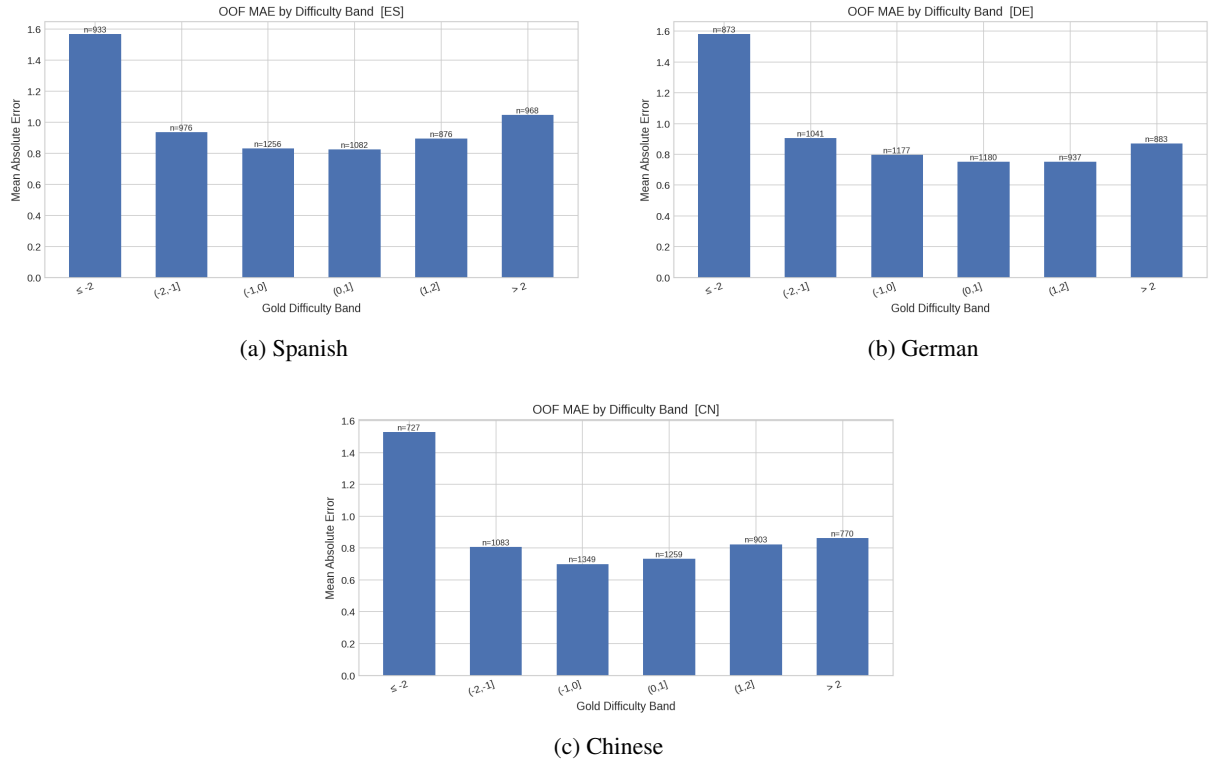


Figure 2: Out-of-fold mean absolute error by gold difficulty band for the three L1 conditions.

conditions without any language-specific tuning. This suggests that the task contains enough shared structure for a single pipeline design to work reliably across Spanish, German, and Chinese. The combination of the target word, spelling clue, L1 translation, and L1 context appears to provide sufficient signal for a multilingual encoder to learn useful patterns of vocabulary difficulty.

Why German benefits most. The largest improvement is observed for German. One possible explanation is that the German subset aligns especially well with the multilingual encoder and the added scalar features. However, this remains a hypothesis rather than a firm conclusion, since verifying it would require a more detailed per-language analysis.

Difficulty extremes are hardest. Figure 2 shows that prediction error is highest for the easiest items and rises again, though less sharply, for the hardest ones. Because the same pattern appears across all three L1 conditions, it is more likely to reflect a general regression-to-the-mean effect than a language-specific issue. This suggests that future work should focus more directly on improving calibration at the tails of the score distribution.

Gap to top-ranked systems. Although our system outperforms the official baseline, it still remains below the top closed-track submissions. The most likely reason is limited model scale: we rely on a single base-sized encoder and a modest 5-fold ensemble under a restricted Kaggle compute budget. Larger multilingual encoders and stronger ensembles would likely improve performance further while still remaining within the closed-track rules.

Reproducibility. A practical strength of the system is that the same notebook is used for all three languages, with only the L1 identifier changed between runs. In addition, each checkpoint stores its own normalization statistics together with the model weights, which makes inference reproducible directly from the saved checkpoints. This keeps the pipeline simple to rerun, inspect, and extend.

8 Conclusion

We presented a unified, language-agnostic closed-track system for the BEA 2026 Shared Task on vocabulary difficulty prediction. Using one notebook, one model, one feature set, and one training recipe across Spanish, German, and Mandarin Chinese,

our system consistently outperforms the official baseline on all three L1 conditions. Error analysis reveals a consistent pattern across languages: prediction error is highest for the easiest vocabulary items, suggesting a systematic regression-to-the-mean tendency that future work could address. We hope this pipeline serves as a practical and accessible starting point for future work on L1-aware vocabulary modeling.

Limitations

Our system does not achieve top-ranked performance. A main reason is compute: under a budget of two NVIDIA T4 GPUs on the Kaggle free tier, we used xlm-roberta-base and a 5-fold ensemble of the same architecture rather than a larger encoder or a multi-encoder ensemble. Such extensions remain compatible with the closed-track rules but were beyond our available resources.

Our feature set is also intentionally minimal. We use only four character-count features, which capture simple surface-level signals but omit potentially useful information such as word frequency, morphological complexity, subword count, and cognate similarity. These additions would still be closed-track compliant and are a natural direction for future work.

Although the pipeline is language-agnostic in design, it has been evaluated only on the three L1 conditions provided by the shared task. Generalization to other L1s therefore remains untested. Finally, our test predictions are based on fold models trained on the training split only; retraining on the combined training and development data might yield further gains. Because cuDNN benchmark mode is enabled for speed, small run-to-run variation across GPU setups is also possible.

Ethical Considerations

Vocabulary difficulty prediction has direct applications in adaptive testing and personalized content selection. However, prediction errors can negatively affect learner experience, particularly for learners at the easiest end of the difficulty distribution, where our analysis shows error is consistently highest across all three L1 conditions (Figure 2). We used only the officially released shared-task data and comply with all closed-track constraints. No personally identifiable learner information is present in the data. As with any automated scoring system, outputs should inform rather than replace

human pedagogical judgment, especially in high-stakes assessment contexts.

Acknowledgements

We used ChatGPT (OpenAI) for code assistance and Claude (Anthropic) for manuscript preparation. No generative AI model was used as part of the submitted closed-track system, data, or prediction pipeline. All results are from the official shared-task leaderboard and were verified by the authors, who take full responsibility for this submission.

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. [Optuna: A next-generation hyperparameter optimization framework](#). In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.
- Mariano Felice and Lucy Skidmore. 2026. BEA 2026 shared task: Vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. Introducing knowledge-based vocabulary lists (KVL). *TESOL Journal*, 12(4).
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. [Knowledge-based Vocabulary Lists](#). British Council.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 160–174. Association for Computational Linguistics.

BoostedCats at BEA 2026 Shared Task 1: What Makes a Word Hard to Learn? Modeling L1 Influence on English Vocabulary Difficulty

Jonas Mayer Martins Zhuojing Huang Aaricia Herygers Lisa Beinborn
University of Göttingen, Germany
firstname.lastname@uni-goettingen.de

Abstract

What makes a word difficult to learn, and how does the difficulty depend on the learner’s native language? We computationally model vocabulary difficulty for English learners whose first language is Spanish, German, or Chinese with gradient-boosted models trained on features related to a word’s familiarity (e.g., frequency), meaning, surface form, and cross-linguistic transfer. Using Shapley values, we determine the importance of each feature group. Word familiarity is the dominant feature group shared by all three languages. However, predictions for Spanish- and German-speaking learners rely additionally on orthographic transfer. This transfer mechanism is unavailable to Chinese learners, whose difficulty is shaped by a combination of familiarity and surface features alone. Our models provide interpretable, L1-tailored difficulty estimates that can be used to design vocabulary curricula.

 [Code repository and interactive demo](#)

1 Introduction

Learning a second language begins with words. Developing an extensive vocabulary is essential for mastering grammar and achieving fluency (Schmitt and Schmitt, 2020; Nation, 2000). Yet not all words are equally hard to learn, and researchers of second-language (L2) acquisition, classroom practitioners, and creators of educational materials seek to better understand this phenomenon.

The difficulty of a word is partially due to factors common to all learners, that is, lexical properties that depend only on the target language itself (Peters, 2019). For example, common and concrete words are easier to acquire than rare and abstract ones (Ellis, 2002; Ellis and Beaton, 1993).

However, a learner’s individual background—their first language (L1), in particular—plays a key role in word difficulty (Ringbom and Jarvis, 2009). For instance, a German speaker readily infers that

the English word “sheep” is related to the cognate *Schaf*, whereas the Spanish *oveja* is not a cognate.¹ This orthographic bridge supports active as well as passive vocabulary competence. Conversely, false friends may lead to interfering transfer (Bensoussan and Laufer, 1984). For example, *bravo* in Spanish is close to its English cognate “brave”, yet *brav* in German means “well-behaved”. The same English word can thus be easy for one learner while difficult for another, depending on the relationship between the L1 and L2.

However, computational models of lexical difficulty often fail to take a learner’s background into account. The SemEval2021 shared task on lexical-complexity prediction, for example, considers difficulty for unspecified annotators (Shardlow et al., 2021). Meanwhile, predictions of lexical difficulty tailored to individual learners are receiving increasing attention (Palmero Aprosio et al., 2020; North et al., 2023; Schmitt et al., 2021; Skidmore et al., 2025), yet studies that do include L1 effects often remain limited to a single language pair.

Approach and contributions. We address this gap by modeling English vocabulary difficulty separately for Spanish, German, and Chinese² L1 speakers, using the Knowledge-based Vocabulary Lists (KVL), a recent crowd-sourced corpus of lexical-difficulty tests (Schmitt et al., 2021; Skidmore et al., 2025), in the context of the closed track of the Building Educational Applications (BEA) 2026 shared task (Felice and Skidmore, 2026). We create four feature groups: the familiarity of a word based on exposure effects, its meaning based on semantic complexity and concreteness, its surface form based on orthographic complexity, and its transferability from a specific L1 based on string similarity. We train interpretable, gradient-boosted

¹Instead, *oveja* is distantly related to “ewe”, meaning female sheep, in English.

²In this article, we refer to Mandarin Chinese as Chinese.

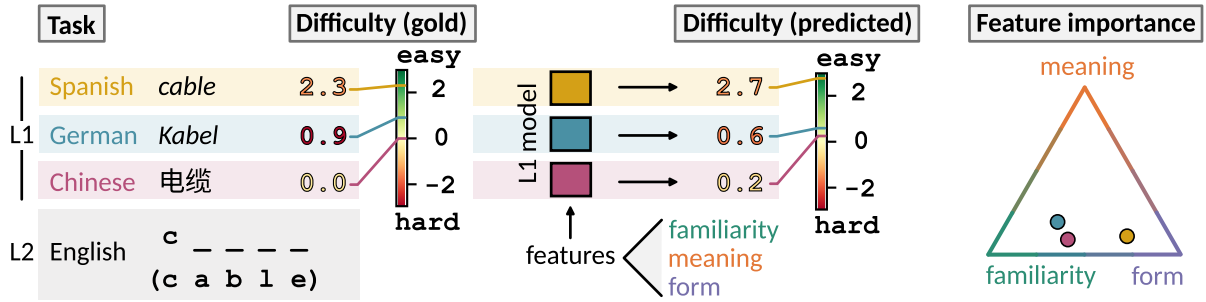


Figure 1: Illustration of the task and modeling setup. For each L1, learners translate a word (e.g., Kabel) from their L1 into English given the first letter (c) and a graphical hint indicating the number of characters. Item difficulty is estimated from aggregated responses (gold label). We train L1-specific models on feature groups (familiarity, meaning, and form = surface $\cup$ transfer). By evaluating the feature importance, we assign a relative importance in the prediction of a word to each feature group.

models on these features to predict L1-dependent lexical difficulty and trace their predictive power via Shapley values. Figure 1 illustrates the task format and our approach.

To showcase our results and support further research and practical use, we have created an [interactive demo](#) that extends our predictions to more than 4,000 to 7,000 additional English words per language, see Appendix A and Appendix B. This demo highlights the advantage of our computational modeling approach, which allows us to make predictions beyond the limited KVL data for a larger range of vocabulary and potentially more L1 backgrounds, providing tailored difficulty estimates applicable to curriculum design.

2 Related work

In this section, we discuss factors of word difficulty in a foreign language based on second-language-learning studies and computational models.

2.1 Lexical difficulty

Vocabulary knowledge poses a fundamental challenge to all second-language (L2) learners (Schmitt and Schmitt, 2020; Nation, 2000). However, the difficulty associated with a particular word varies by learner. Some words, for instance, represent intrinsically difficult concepts in the target language, while others are challenging or easy specifically due to a learner’s L1 background.

L2-intrinsic difficulty. The more frequent a stimulus occurs, the more readily it tends to be recalled, and as such, frequent words are learned more easily because learners encounter them more often (Ellis, 2002; Peters, 2019). Educational curricula typically introduce frequent words first to enable learn-

ers to speak and understand the target language earlier by efficiently building vocabulary coverage (Nation, 2000).

Beyond frequency, other lexical properties are also important. For instance, orthographic complexity has been shown to predict exercise difficulty (Beinborn et al., 2016). Additionally, homonymous or polysemous words are typically harder to process and acquire than words with just one meaning (Bensoussan and Laufer, 1984). Psycholinguistic features, such as the age of acquisition (Kuperman et al., 2012; Dascalu et al., 2016) and concreteness of the word (Ellis and Beaton, 1993; Van Hell and Mahn, 1997; De Groot and Keijzer, 2000), are important factors for word difficulty prediction, too. The initial letter plays a disproportionate role in word recognition and production: speakers in tip-of-the-tongue states often remember the first letter and phonological primes aid retrieval (Brown and McNeill, 1966; James and Burke, 2000).

L1-specific difficulty. Beyond L2-intrinsic word properties, difficulty is shaped by the learner’s first language. Learners draw on L1 knowledge when acquiring an L2 vocabulary, with positive transfer where the languages are similar and interference where surface similarity is misleading (Odlin, 1989). At the lexical level, cognateness is a well-studied transfer mechanism. Cognates reduce learning effort because learners can map the L2 target onto the well-known L1 form: De Groot and Keijzer (2000) show that cognateness predicts L2-vocabulary retention, and Beinborn et al. (2014) find that learners infer the meaning of unfamiliar words more readily when there is a cognate in their L1. Similarly, Urdaniz and Skoufaki (2022) find that frequency and cognateness are the two

strongest predictors of lexical difficulty for Spanish learners of English, with significant interaction such that word frequency is more predictive for non-cognates than cognates. However, coincidental similarity or shifted meaning of cognates can cause interference (Odlin, 1989; Bensoussan and Laufer, 1984).

These findings imply that the same word can be easy for one learner and hard for another, depending on the overlap of L1 and L2. Moreover, cognateness interacts with other predictors of difficulty, motivating models that jointly incorporate L1-specific and L2-intrinsic information.

2.2 Predicting lexical difficulty

Progress on predicting lexical difficulty³ has been driven by shared-task competitions (Paetzold and Specia, 2016; Yimam et al., 2018; Shardlow et al., 2021). Computational approaches to predicting lexical difficulty have shifted from binary classification of complex words to continuous regression of difficulty scores (North and Zampieri, 2023). Across studies, robust predictors include word frequency, word length, age of acquisition, and concreteness (North et al., 2023).

These patterns extend beyond English. The CWI-2018 shared task shows that models trained on English, German, and Spanish can generalize to French, suggesting that certain cognitive mechanisms transfer across typologically related languages (Yimam et al., 2018). The BEA 2024 shared task investigated lexical-complexity prediction and simplification with 10 languages, finding that feature-based systems can compete with large language models (Shardlow et al., 2024). For some languages, script-specific features are important, e.g., logographic-character frequency in Chinese and Japanese (Lee and Yeung, 2018a; Nishihara and Kajiwara, 2020). Furthermore, Finnimore et al. (2019) confirm features from typologically related languages can be beneficial, while including features from unrelated languages may reduce performance.

Despite these advances, learner-specific factors remain underexplored. Studies that model the L1 background report benefits over agnostic models, e.g., for Dutch learners of French as well as Chinese and Spanish learners of English (Tack et al.,

2016; Lee and Yeung, 2018b; Urdaniz and Skoufaki, 2022). However, these studies focus on a single language pair. Palmero Aprosio et al. (2020) extend this line of research to multiple L1s for learners of Italian by modeling cognates and false friends. Similarly, the lexical complexity for learners of Japanese has been studied for several L1 backgrounds (Ide et al., 2023; Nohejl et al., 2024). Still, the interaction between L1-specific and general features remains largely unexamined. In this article, we address that gap directly.

3 Experimental setup

We train and evaluate L1-specific CatBoost models (Prokhorenkova et al., 2018) for predicting the difficulty of English vocabulary items. This section describes the dataset, feature groups, model, and evaluation metrics.

3.1 Data

We use the Knowledge-based Vocabulary Lists (KVL), a large-scale dataset of lexical difficulty based on vocabulary test responses from more than 100,000 second-language learners of English (Schmitt et al., 2021; Skidmore et al., 2025) as part of the BEA 2026 shared task (Felice and Skidmore, 2026). The data covers three L1s—Spanish, German, and Chinese—with 6,091 training, 677 development, and 748 test items per L1.

In the KVL test format, learners see a word in their L1 with a context sentence and must type the English translation, given the first letter and blanks for the remaining letters:

Kabel German, noun
Achtung, stolpere nicht über das Kabel am Boden.
c _ _ _ _

Each item is scored as correct or incorrect. In the example above, the correct completion would be *able*. Responses are aggregated into a continuous logarithmic difficulty score by a Rasch model (De Boeck, 2008; Dunn, 2024), which jointly estimates item difficulty and learner ability such that the resulting centered score reflects the intrinsic difficulty of a word (higher scores are easier).⁴

The difficulty estimates may vary by L1, see Fig. 1. The word *cable* is easiest for Spanish speakers (score 2.3) because the corresponding

³In this article, we use *difficulty* to refer to the processing effort in retrieval, as defined in Bulté et al. (2025), and distinguish this from structural complexity, which concerns the internal composition of a word.

⁴Note that the values across languages are not directly comparable in an absolute way but rather as relative to the average word difficulty of an L1.

Spanish form is spelled identically; moderately easy for German speakers (score 0.9), who benefit from the German cognate *Kabel*; and hardest for Chinese speakers (score 0.0, 电缆 *diànlǎn*).

3.2 Features

We extract 24 features per item and organize them into four feature groups that structure our analysis, see Fig. 1. Each group targets a different aspect of what makes a word easy or difficult to recall. Full definitions and sources are provided in Appendix C. Here, we describe the rationale behind each group.

Familiarity (11 features). These features capture how likely a learner is to have encountered a word before. We include logarithmic word frequency and contextual diversity from SUBTLEX-UK (Van Heuven et al., 2014), reported age of acquisition and percentage of annotators who knew the word (Kuperman et al., 2012), CEFR level in the form of CEFR-J proficiency level, e.g., B2 for cable, and the EFLLex word-frequency profile across CEFR-graded books (Europarat, 2011; Negishi et al., 2013; Dürlich and François, 2018). Together, these features reflect both naturalistic exposure (how often the word appears in English media) and curricular sequencing (at which proficiency level the word is typically introduced).

Meaning (5 features). To approximate semantic complexity and concreteness, we use the ℓ_2 norm of the English fastText embedding (Bojanowski et al., 2017), mean hypernym depth and sense count from WordNet⁵ (Fellbaum, 1998), the part-of-speech (POS) ratio from SUBTLEX-UK, and a binary flag indicating whether the KVL item required disambiguation, e.g., for the item decode, the German translation is *etw entschlüsseln* (nicht: decipher).

Surface (7 features). Surface features describe the orthographic form of the item as presented in the test. They include target- and source-word length in characters, number of syllables and letters per phoneme (an orthographic transparency proxy) of the target word, context-sentence length, and the first letter of both the English clue and the L1 translation. These features are relevant because the KVL task requires learners to reconstruct a spelling, making word length, letter cues, and phoneme den-

sity directly task-relevant (James and Burke, 2000; Beinborn et al., 2016).

Transfer (1 feature). We compute the cosine similarity between character n -gram TF-IDF vectors of the English word and its L1 translation, see Appendix D for a detailed description. This feature captures orthographic overlap as a proxy for cognateness: High values indicate transparent cognates (e.g., English *fantasy* and Spanish *fantasía*), while the similarity is zero for all Chinese items due to the different script. Although this measure does not account for phonological similarity or regular sound correspondences, we find it to be highly informative for languages that share the Latin script with English.

3.3 Model and baselines

For our purposes, we require a fast and interpretable model. We use CatBoost (Prokhorenkova et al., 2018), a gradient-boosted decision-tree method that handles both numeric and categorical features. We train one model per L1 on the same set of 24 features (Fig. 1, right), allowing each model to learn L1-specific interactions between features and difficulty. Hyperparameters are given in Appendix E. Model training takes about ten seconds. To estimate variance, we train each model with 20 random seeds.

We compare against two baselines: a ridge regression model trained on the same features, which tests whether nonlinearity or feature interactions contribute beyond a linear combination, and the transformer-based approach of Skidmore et al. (2025), which fine-tunes a pretrained language model directly on KVL difficulty scores without handcrafted features.

3.4 Evaluation

We report the root-mean-square error (RMSE), in alignment with the BEA 2026 shared task (Felice and Skidmore, 2026), and Pearson’s r on the test set. Our team’s submission results (*Boosted Cats – HuDS lab*) are available at the [shared-task repository](#). To test whether L1-specific models capture shared or distinct difficulty structures, we additionally evaluate each model on the test sets of the other two L1s.

For interpretability, we use tree-based SHAP values (Lundberg and Lee, 2017; Lundberg et al., 2018), which attribute each prediction to individual feature contributions through Shapley values from

⁵For example, the word *cable* has six senses, one of which has a hypernym path of depth 9: *cable* $\subseteq$ *conductor* $\subseteq$ *device* $\subseteq$ *instrumentality* $\subseteq$ *artifact* $\subseteq$ *whole* $\subseteq$ *object* $\subseteq$ *physical entity* $\subseteq$ *entity*.

cooperative game theory (Shapley, 1953). We aggregate absolute Shapley values by feature group to obtain their relative importance, analyzed in Section 4.2. We compute relative group importance as the sum of absolute Shapley values within each feature group, normalized to sum to one per item.

4 Results

The same English word can be easy for one L1 group and hard for another, illustrated by the gold-label lexical difficulties that correlate across L1s but are far from identical, see Appendix F. This observation motivates our approach to train L1-specific models, asking three questions. Are the model predictions accurate enough to warrant a study of how these predictions arise (Section 4.1)? Which mechanisms do the models rely on, and how do these mechanisms differ by L1 (Section 4.2)? And finally, do models trained on one L1 generalize to another (Section 4.3)?

4.1 Prediction quality

Table 1 compares our CatBoost models against two baselines: a linear (ridge) regression trained on the same features and the transformer-based approach of Skidmore et al. (2025). CatBoost achieves the lowest RMSE on all three L1s and the highest correlation in German and Chinese. The improvement over the linear baseline shows the importance of accounting for feature interactions and non-linear contributions to difficulty. CatBoost outperforms the transformer baseline, too, implying that well-chosen psycholinguistic features can perform on par with pretrained representations while being more easily interpretable.

Model	RMSE ↓			Pearson’s r ↑		
	ES	DE	CN	ES	DE	CN
Transformer	1.26	1.26	1.14	0.77	0.77	0.75
Linear regression	1.30	1.20	1.07	0.72	0.74	0.77
CatBoost (ours)	1.24	1.12	1.04	0.76	0.78	0.79

Table 1: Test-set performance per L1. CatBoost results report the median over 20 random seeds; the 5-95% range is ≤ 0.02 for all entries. Best result per L1 in **bold**. The shared-task baseline fine-tunes a transformer on KVL difficulty scores (Skidmore et al., 2025).

Figure 2 shows predictions against the gold labels per L1. The predictions correlate well with the gold labels, as the bulk of items clusters near the diagonal (dashed gray). The hardest items, however, tend to be predicted as easier than they

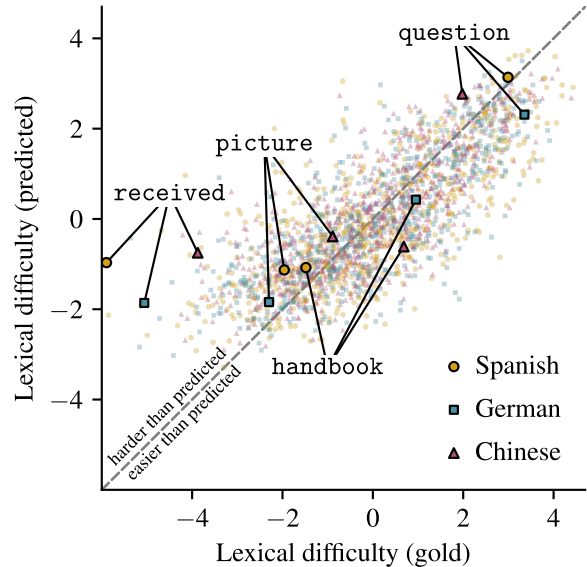


Figure 2: Predicted versus gold-label lexical difficulty. Each point is one L1-English vocabulary pair; the dashed diagonal marks a perfect prediction. Four exemplary items are highlighted: question (noun), picture (verb), handbook (noun), received (adjective). Seed variation of the prediction (5th–95th percentile) is small (median 0.1, max. < 0.4).

are. This bias is consistent across L1s, suggesting that the most difficult words involve challenges not captured entirely by our feature set, e.g., rare senses (received in the meaning of “generally accepted”). Other highlighted examples include question, which is easy and well-predicted for all L1s, although it is a cognate only in Spanish; to picture and handbook show that our model captures the L1-dependent difficulty.

4.2 Feature-group importance

To understand the model predictions more intuitively, we thus group our 24 features into four feature groups—*familiarity*, *meaning*, *surface*, and *transfer*—and aggregate the absolute Shapley values per item for each group. The importance share of a feature group measures how strongly the features within this group contribute to predicting a specific item. We report the average importance of every feature and its correlation with the gold-label difficulty in Table 2.

Transfer differentiates Spanish and German from Chinese learners. The most prominent cross-linguistic difference is the role of transfer. For Spanish and German speakers, a single feature—character- n -gram cosine similarity between the L1 and the English word—is the

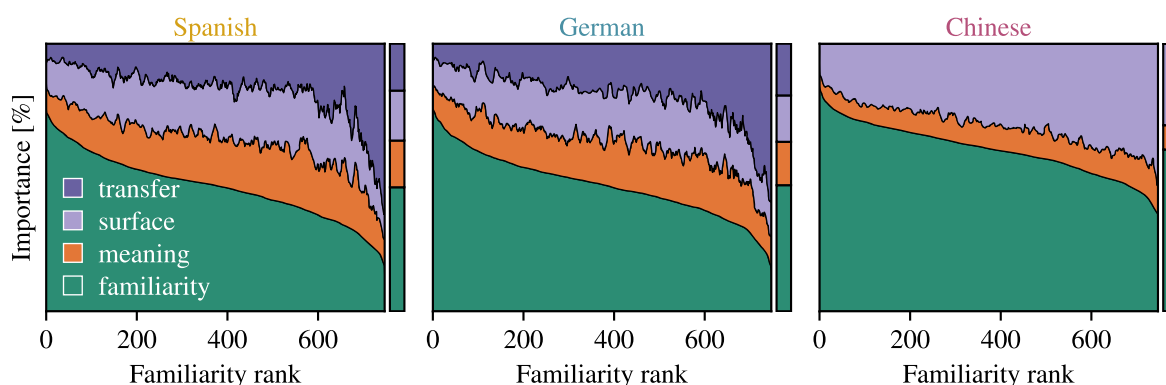


Figure 3: Per-item feature-group-importance shares, sorted by decreasing importance of familiarity (left to right). Median across 20 seeds, with a rolling average of window size 10. The bar at the right of each panel shows for each group the median across seeds of its mean importance share over all item. The *form* group is split into *transfer* (character similarity between the L1 and English) and *surface* (features like word length and initial letters).

strongest predictor overall (mean $|\text{SHAP}|$ of 0.51 for Spanish and 0.52 for German; see Fig. 8 in Appendix C). For Chinese, this feature does not contribute at all because the Chinese writing system shares no characters with English.

This asymmetry is visible in Fig. 3, which shows the group importance of each item, sorted by decreasing importance of familiarity. Transfer fills the gap of less familiar words for Spanish and German. These are typically cognates, e.g., *handbook* and *Handbuch*, which a German speaker can infer even without prior exposure to the English word.⁶ Transfer is most relevant for a distinct small subset of roughly 10–20% of items relying least on familiarity (above rank 600).

Familiarity is strong across L1s. Familiarity accounts for the largest average feature-group importance for all three L1s, see the stacked bar summaries in Fig. 3. Within this group, the most important features differ: age of acquisition (AoA) and frequency for Spanish and German, versus the proficiency levels (EFLLex-level span and CEFR-J levels) for Chinese. This difference may reflect a variation in how English is acquired—naturalistic exposure for Spanish and German learners as opposed to potentially more structured, curriculum-driven learning for Chinese learners. Yet in all cases, familiar words are easier to recall.

Surface and meaning. Surface features are as important as meaning features (embedding norm, hypernym depth, and sense count) for Spanish and

German, and substantially more important for Chinese, see Fig. 3. The importance of surface features is likely due to the KVL task format, which requires form recall rather than correct usage of a word. Spelling difficulty makes surface cues, e.g., word length and priming by the clue letter, directly task-relevant (James and Burke, 2000; Beinborn et al., 2016). Chinese learners appear to rely on these cues more heavily.

Two routes to easiness. Figure 4 projects each item onto a triangle. Each corner represents that the respective feature group is uniquely important, while the middle corresponds to an equal mixture of importance. The color of each point encodes gold-label difficulty and the background shading is an estimate of the difficulty distribution in this triangle.⁷

For Spanish and German, easy items (green) cluster in the two lower corners, familiarity and form. These clusters represent two routes through which a word is easy to a learner. A word is easy because it is frequent and early-acquired or because it closely resembles its L1 form. Some words are both common and have high orthographic similarity, which makes them easy as well (see Appendix G for details), but they land in the middle region since the Shapley values for familiarity and transfer are both high. Difficult items (red) occupy the middle region in the triangle because neither cue is strong.

For Chinese, the distribution in the triangle is instead tightly clustered along the lower-left

⁶Transfer beyond orthographic overlap exists but is not captured by character similarity. For example, the Chinese word 手册 (*shǒucè*, literally “hand-book”) elicits transfer, too.

⁷A Nadaraya–Watson kernel regression of the difficulty surface (Nadaraya, 1964; Watson, 1964) with bandwidth selected by leave-one-out cross-validation (Härdle, 1990).

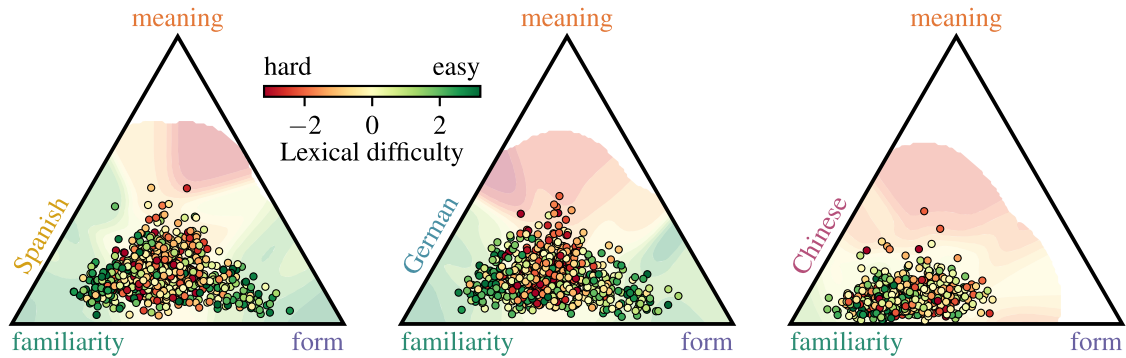


Figure 4: Each item projected onto a triangle according to the relative importance of three feature groups (familiarity, meaning, and form = surface $\cup$ transfer). Color encodes the gold-label difficulty of an item (red: hard, green: easy). Background shading shows a regression of the difficulty surface; regions with insufficient data are masked.

familiarity-form edge⁸ and the background shading has no clear gradient. Chinese difficulty is shaped more uniformly by a combination of familiarity and form features.

4.3 Cross-L1 transfer of models

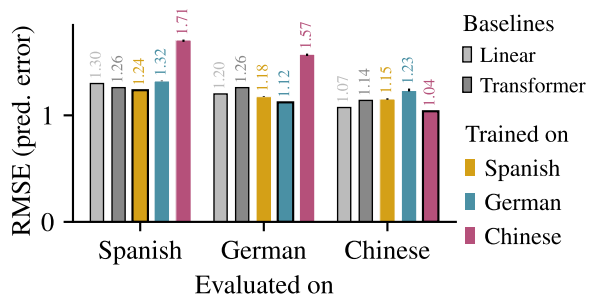


Figure 5: Cross-L1 evaluation. Each colored bar shows the RMSE prediction error (lower is better) of a CatBoost model trained on one L1 and evaluated on the L1 indicated on the horizontal axis. Thick black outlines mark within-L1 evaluation. Gray bars show the ridge linear baseline and the transformer baseline, each trained on the target L1. CatBoost results report the median over 20 random seeds with 5th–95th percentiles.

If Spanish and German learners rely on similar feature groups, a model trained on one L1 should reasonably generalize to the other. We test this by evaluating each L1-specific CatBoost model on all three test sets, see Fig. 5.

Each model performs best on its own L1, but the degradation when transferring between Spanish and German is minor. The Spanish-trained model achieves RMSE 1.18 on German (vs. 1.12 within-German), and the German-trained model

reaches RMSE 1.32 on Spanish (vs. 1.24 within-Spanish)—both on par with the linear baselines (RMSE 1.30 for Spanish, 1.20 for German) and the transformer baselines (RMSE 1.26 for Spanish, 1.26 for German) trained directly on the target L1. A model trained on Spanish or German thus captures meaningful mechanisms that transfer to the other, without any access to target-L1 training data.

Transfer involving Chinese is markedly worse in both directions. Chinese-trained models yield RMSE 1.71 on Spanish and 1.57 on German, well above the baselines. Spanish- and German-trained models evaluated on Chinese (RMSE 1.15 and 1.23) degrade less severely, which aligns with the observation that the Spanish and German models have learned all the features relevant to predicting difficulty for Chinese learners, although they still fall short of the within-Chinese model (RMSE 1.04). The Chinese-trained model, not having learned to predict items based on transfer, cannot generalize to Spanish or German.

These cross-L1 results mirror the feature-group analysis: Spanish and German models share a difficulty landscape, while the prediction for Chinese learners relies on different cues. Practically, this observation suggests that models trained on a given L1, say Spanish, could provide useful estimates for typologically related languages such as Portuguese or Italian, even without the dedicated training data.

5 Discussion

Our finding that word familiarity exerts the strongest impact on the KVL difficulty scores across all L1 groups aligns with prior research identifying word frequency and age of acquisition as robust predictors of lexical processing (Rott, 1999;

⁸Chinese items show a somewhat lower dispersion in the feature-group simplex (Aitchison total variance: CN 0.30, ES 0.33, DE 0.35 with bootstrap 90% CIs < 0.05).

Ellis, 2002; Kuperman et al., 2012; North et al., 2023). Our analysis adds a quantitative decomposition that reveals *how* this shared component interacts with L1-specific factors. For Spanish and German, transfer provides a second route to easiness: Words can be easy because they are familiar or because they are orthographic cognates. This two-route structure aligns with Urdaniz and Skoufaki (2022), who find frequency and cognateness to be strong, interacting predictors for Spanish learners of English, and with De Groot and Keijzer (2000), who show that cognates are easier to remember. We observe the same pattern for German learners. For Chinese learners, difficulty is predicted more uniformly by surface features and familiarity. The cognitive strategies during form recall thus appear to differ qualitatively, depending on the typological relationship between the L1 and L2.

Our character similarity feature captures character overlap as a proxy for perceived similarity (Ringbom, 1987), which need not coincide with etymological relatedness. The strength of this feature in our models suggests that even coarse character overlap is a powerful retrieval cue when L1 and L2 share a script. Although Chinese has no orthographic overlap, cross-linguistic transfer may still be present in the form of loanwords (e.g., 雷达 *lédá* from English radar), calques (brainwash from the Chinese 洗脑 *xǐnǎo*), and parallel compositions (星光 *xīngguāng* happening to map to “starlight”) or morphology (地 *de* marking adverbs like the English suffix “-ly”).

The task format affects which aspects of word knowledge are tested (Laufer and Goldstein, 2004; Culligan, 2015). In our case, meaning features contribute little across all L1s, likely because the KVL task tests spelling rather than an understanding of meaning. A task requiring contextual usage would likely involve more meaning-related features.

A caveat concerns interpreting Shapley values as cognitively plausible. Our models approximate human test responses, and their strong held-out performance suggests that the features are plausible factors in the learner behavior that underlies the test responses. Where the model fails, however, the relevant factors likely lie outside our feature set, e.g., individual learner differences, contextual effects of the context sentence, or interference from false friends. We discuss the relation of lexical features beyond our predictor set with the gold-label difficulty and model-prediction errors in Appendix H.

Our findings also have practical implications.

Spanish and German learners could benefit from curricula rich in cognates in order to efficiently build an extensive vocabulary (Nation, 2000). Furthermore, a model trained on data from one L1 could serve in predicting lexical difficulty for typologically related, lower-resource languages without dedicated data. For Chinese learners of English (and potentially speakers of languages not related to the target language), a curriculum based on frequency and proficiency levels is likely more effective. Since our model can generalize to unseen words, we provide predictions of lexical difficulty beyond the KVL dataset, on the entire SUBTLEX-UK vocabulary for which we could compute the features. Our model predictions can be explored [interactively](#).

6 Conclusions

We have investigated what makes an English word difficult for second-language learners by training interpretable models on lexical-difficulty scores from Spanish, German, and Chinese L1 speakers. By decomposing the predictions into feature groups, we identify two qualitatively different profiles. Spanish and German learners benefit from two routes to easiness—familiarity and orthographic transfer—while Chinese learners rely on a less structured combination of familiarity and surface features in the absence of orthographic overlap. The models of learners with Spanish and German as L1 generalize well to each other, indicating that the learned difficulty functions reflect shared cognitive strategies.

For future research, the strong cross-L1 transfer between the Spanish- and German-trained models suggests that models trained on a high-resource L1 could provide useful difficulty estimates for typologically related languages (e.g., from Spanish to Portuguese, Catalan, or Italian; potentially also from Chinese to other languages not related to English like Japanese), even without the dedicated training data. Another promising direction is to investigate whether the identified mechanisms are symmetric, i.e., whether the same profile of feature-group importance applies for English speakers that learn Spanish, German, or Chinese. Third, incorporating phonological similarity and including calques and morphological similarity could capture part of the variance in lexical difficulty that our current feature set does not explain.

Limitations

Language coverage. The KVL data covers a limited selection of four high-resource languages, three of which are Indo-European. The extent to which the patterns we identify generalize to learners with other language backgrounds remains an open question.

Task specificity. The KVL dataset tests productive form recall only. Our analysis therefore pertains to this aspect of vocabulary knowledge and could be extended to other test formats.

Measuring transfer. We operationalize cross-linguistic transfer through character-level overlap, which does not capture phonological similarity, regular sound correspondences, and semantic transfer (e.g., calques).

Interpretation of feature importance. Feature-group importance describes the structure of model predictions and cannot measure a learner's cognitive processes directly. Inferences about human cognition thus remain indirect.

Curricular effects. Lexical difficulty is not only influenced by word properties but also the curriculum order, which is captured by our CEFR-proficiency-level features. Beyond the intrinsic difficulty of a word, the KVL difficulty we model therefore partially encodes the status quo in instruction regarding where a word happens to be placed in educational curricula.

Ethical considerations

This work is intended as fundamental research into L2 vocabulary difficulty. While our results may inform applications and further research, it should not be interpreted as prescriptive guidance for teaching in practice.

Our computational experiments use light-weight models that entail no substantial environmental impact.

Acknowledgments

We thank the reviewers for their thoughtful and detailed feedback. This research is partially supported by the zukunft.niedersachsen program of the VolkswagenStiftung (L.B., Z.H.) and by a VENI grant (VI.Veni.211C.039) from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) (L.B., A.H.).

Author contributions

J.M.M.: Conceptualization, Analysis, Software, Literature review, Visualization, Writing—original draft, Writing—review & editing. **Z.H.:** Analysis and Software (feature analysis for App. H, KVL extension), Literature review, Visualization (App. H), Writing—original draft (App. B, D, and H; parts of Introduction and Related Work), Writing—review & editing. **A.H.:** Literature review, Writing—original draft (parts of Related Work). **L.B.:** Supervision, Conceptualization, Analysis, Literature review, Writing—review & editing.

References

- Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. [Readability for foreign language learning: The importance of cognates](#). *ITL - International Journal of Applied Linguistics*, 165(2):136–162.
- Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2016. [Predicting the spelling difficulty of words for language learners](#). In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 73–83, San Diego, CA, USA. Association for Computational Linguistics.
- Marsha Bensoussan and Batia Laufer. 1984. [Lexical guessing in context in EFL reading comprehension](#). *Journal of Research in Reading*, 7(1):15–32.
- 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.
- Roger Brown and David McNeill. 1966. [The “tip of the tongue” phenomenon](#). *Journal of Verbal Learning and Verbal Behavior*, 5(4):325–337.
- Bram Bulté, Alex Housen, and Gabriele Pallotti. 2025. [Complexity and difficulty in second language acquisition: A theoretical and methodological overview](#). *Language Learning*, 75(2):533–574.
- Brent Culligan. 2015. [A comparison of three test formats to assess word difficulty](#). *Language Testing*, 32(4):503–520.
- Mihai Dascalu, Danielle McNamara, Scott Crossley, and Stefan Trausan-Matu. 2016. [Age of exposure: A model of word learning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, Phoenix, AZ, USA. AAAI Press.
- Paul De Boeck. 2008. [Random item IRT models](#). *Psychometrika*, 73(4):533–559.
- Annette M. B. De Groot and Rineke Keijzer. 2000. [What is hard to learn is easy to forget: The roles of word concreteness, cognate status, and word frequency in foreign-language vocabulary learning and forgetting](#). *Language Learning*, 50(1):1–56.

- Karen J. Dunn. 2024. [Random-item rasch models and explanatory extensions: A worked example using L2 vocabulary test item responses](#). *Research Methods in Applied Linguistics*, 3(3):100143.
- Luise Dürlich and Thomas François. 2018. [EFLLex: A graded lexical resource for learners of English as a foreign language](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- 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.
- Nick C. Ellis and Alan Beaton. 1993. [Psycholinguistic determinants of foreign language vocabulary learning](#). *Language Learning*, 43(4):559–617.
- Europarat, editor. 2011. *Common European framework of reference for languages: Learning, teaching, assessment*, 12th edition. Cambridge University Press, Cambridge, UK.
- Mariano Felice and Lucy Skidmore. 2026. Shared task on vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, CA, USA. Association for Computational Linguistics.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*, 1st edition. The MIT Press, Cambridge, MA, USA.
- Pierre Finnimore, Elisabeth Fritsch, Daniel King, Alison Sneyd, Aneeq Ur Rehman, Fernando Alva-Manchego, and Andreas Vlachos. 2019. [Strong base-lines for complex word identification across multiple languages](#). In *Proceedings of the 2019 Conference of the North*, pages 970–977, Minneapolis, MN, USA. Association for Computational Linguistics.
- Wolfgang Härdle. 1990. *Applied Nonparametric Regression*, 1st edition. Cambridge University Press, Cambridge, UK.
- 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.
- Lori E. James and Deborah M. Burke. 2000. [Phonological priming effects on word retrieval and tip-of-the-tongue experiences in young and older adults](#). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(6):1378–1391.
- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. [Age-of-acquisition ratings for 30,000 English words](#). *Behavior Research Methods*, 44(4):978–990.
- Batia Laufer and Zahava Goldstein. 2004. [Testing vocabulary knowledge: Size, strength, and computer adaptiveness](#). *Language Learning*, 54(3):399–436.
- John Lee and Chak Yan Yeung. 2018a. [Automatic prediction of vocabulary knowledge for learners of Chinese as a foreign language](#). In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–4, Algiers. IEEE.
- John Lee and Chak Yan Yeung. 2018b. [Personalizing lexical simplification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, NM, USA. Association for Computational Linguistics.
- Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. 2018. [Consistent individualized feature attribution for tree ensembles](#). *arXiv preprint*.
- Scott M. Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777, Long Beach, CA, USA. Curran Associates Inc.
- George A. Miller. 1995. [WordNet: A lexical database for English](#). *Communications of the ACM*, 38(11):39–41.
- Ëlizbar A. Nadaraya. 1964. [On estimating regression](#). *Theory of Probability and Its Applications*, 9(1):141–142.
- Ian Stephen Paul Nation. 2000. *Learning Vocabulary in Another Language*, 1st edition. Cambridge University Press, Cambridge, UK.
- Masashi Negishi, Tomoko Takada, and Yukio Tono. 2013. [A progress report on the development of the CEFR-J](#). In Evelina D. Galaczi and Cyril J. Weir, editors, *Exploring language frameworks: Proceedings of the ALTE Kraków Conference, July 2011*, 1st edition, number 36 in Studies in language testing. Cambridge University Press, Cambridge.
- Daiki Nishihara and Tomoyuki Kajiwarara. 2020. [Word complexity estimation for Japanese lexical simplification](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3114–3120, Marseille, France. European Language Resources Association.
- Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2024. [Difficult for whom? A study of Japanese lexical complexity](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 69–81, Miami, FL, USA. Association for Computational Linguistics.
- Kai North and Marcos Zampieri. 2023. [Features of lexical complexity: insights from L1 and L2 speakers](#). *Frontiers in Artificial Intelligence*, 6:1236963.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. [Lexical complexity prediction: An overview](#). *ACM Computing Surveys*, 55(9):1–42.

- Terence Odlin. 1989. *Language Transfer: Cross-Linguistic Influence in Language Learning*, 1st edition. Cambridge University Press, Cambridge, UK.
- Momose Oyama, Sho Yokoi, and Hidetoshi Shimodaira. 2023. *Norm of word embedding encodes information gain*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2108–2130, Singapore. Association for Computational Linguistics.
- 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, CA, USA. Association for Computational Linguistics.
- Alessio Palmero Aproso, Stefano Menini, and Sara Tonelli. 2020. *Adaptive complex word identification through false friend detection*. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 192–200, Genoa Italy. ACM.
- Elke Peters. 2019. *Factors affecting the learning of single-word items*. In Stuart Webb, editor, *The Routledge Handbook of Vocabulary Studies*, 1st edition, Routledge handbooks in linguistics, pages 125–142. Routledge, Taylor & Francis Group, London, UK.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. *CatBoost: Unbiased boosting with categorical features*. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, pages 6639–6649, Montréal, Canada. Curran Associates Inc.
- Real Academia Española. 2025. *Corpus del Español del Siglo XXI (CORPES)*.
- Håkan Ringbom. 1987. *The Role of the First Language in Foreign Language Learning*, 1st edition. Number 34 in Multilingual matters. Multilingual Matters, Clevedon, UK.
- Håkan Ringbom and Scott Jarvis. 2009. *The importance of cross-linguistic similarity in foreign language learning*. In Michael H. Long and Catherine J. Doughty, editors, *The Handbook of Language Teaching*, 1st edition. Wiley, Clevedon, UK.
- Susanne Rott. 1999. *The effect of exposure frequency on intermediate language learners’ incidental vocabulary acquisition and retention through reading*. *Studies in Second Language Acquisition*, 21(4):589–619.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. *Introducing knowledge-based vocabulary lists (KVL)*. *TESOL Journal*, 12(4):e622.
- Norbert Schmitt and Diane Schmitt. 2020. *Vocabulary in Language Teaching*, 2nd edition. Cambridge University Press, Cambridge, UK.
- Lloyd S. Shapley. 1953. *A Value for n-Person Games*. In Harold William Kuhn and Albert William Tucker, editors, *Contributions to the Theory of Games (AM-28), Volume II*, pages 307–318. Princeton University Press.
- 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 et al. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. *Transformer architectures for vocabulary test item difficulty prediction*. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédric Fairon. 2016. *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*, Paris, France. AFCEP - ATALA.
- Raquel Perez Urdaniz and Sophia Skoufaki. 2022. *Spanish L1 EFL learners’ recognition knowledge of English academic vocabulary: The role of cognateness, word frequency and length*. *Applied Linguistics Review*, 13(4):661–703.
- Janet G. Van Hell and Andrea Candia Mahn. 1997. *Keyword mnemonics versus rote rehearsal: Learning concrete and abstract foreign words by experienced and inexperienced learners*. *Language Learning*, 47(3):507–546.
- Walter J. B. Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. *Subtlex-UK: A new and improved word frequency database for British English*. *Quarterly Journal of Experimental Psychology*, 67(6):1176–1190.
- Geoffrey S Watson. 1964. *Smooth regression analysis*. *Sankhyā: The Indian Journal of Statistics, Series A*, 26(4):359–372.
- 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, LA, USA. Association for Computational Linguistics.

Tatu Ylonen. 2022. [Wiktextextract: Wiktionary as machine-readable structured data](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.

A Demo

We provide an [interactive demo](#) that allows users to explore the KVL dataset and our model predictions, see Fig. 6. Given an input text, words are highlighted according to their predicted or gold-label lexical difficulty. Selecting a word opens a panel showing its lexical difficulty, feature-group importance, top individual features, clue letter and context clue.

To map a word form in the input text to KVL lemmas, we precompute a lookup table of inflected forms using the PYINFLECT library. Because the demo is implemented as a static web application, it does not perform syntactic parsing on the fly. Instead, the most frequent part of speech is shown by default, with the option to select any alternatives. The interface also allows users to switch the language background and optionally include our extended vocabulary (see Appendix B).

B KVL extension

To support lexical-difficulty predictions beyond the original KVL inventory, we construct an extended vocabulary in the KVL task format derived from WIKTIONARY data.

Source and filtering. We begin with Wiktionary JSONL extracts produced by WIKTEXTTRACT (Ylonen, 2022). From these, we retain only English headwords (`lang_code = en`) that appear in SUBTLEX-UK (Van Heuven et al., 2014). This filtering restricts the extension to commonly used words. We further discard entries that are empty, single-character, or all-uppercase (e.g., acronyms). Language-specific filters remove entries that primarily encode inflectional variants, spelling alternatives, or other metalinguistic notes.

Normalization. The retained entries are converted into a compact, standardized representation. From each Wiktionary entry, we extract only the information necessary for the KVL format: the English target word and part of speech, a masked spelling clue, an L1 translation, and an optional L1 context sentence. Duplicate English headwords are removed by keeping the first entry with a valid part-of-speech annotation.

L1-specific processing. For German and Spanish, the L1 source word is taken from the first available translation or, if unavailable, from the shortest gloss phrase. For Chinese, if no translation is available, the L1 source word is extracted from gloss blocks and normalized, including conversion to simplified characters. Context sentences are derived from example translations.

Excluding overlap. To ensure that the extended vocabulary contains no duplicates with the original KVL data, we exclude all English target words that occur in the KVL training, development, or test splits for the corresponding L1. The final output consists of three L1-specific CSV files that follow the original KVL schema and add 7,434 items for Chinese, 4,325 for German, and 6,606 for Spanish.

C Features

Table 2 lists the 24 features that we use for our models, grouped by mechanism. For each feature, we report the median absolute Shapley value per L1, computed by averaging absolute Shapley values over test-set items and then taking the median across 20 random seeds. We additionally report Spearman’s rank correlation between the feature value and gold-label difficulty over test-set items.

Intuitively, the Shapley values quantify how important a feature is on average for the model predictions. Spearman’s rank correlations show how monotonic the relation is between the feature value and lexical difficulty. Particularly interesting are those features for which the two measures do not align. For instance, character similarity has a weak correlation with difficulty for Spanish ($\rho = .10$) and German ($\rho = .25$), yet it is the feature with the highest Shapley-value importance for these two languages. This is in part because transparent cognates are relatively rare, and similarity is high only for this subset of transparent cognates. We discuss this in Appendix D. Moreover, the character similarity, as we show in our main analysis Section 4.2, interacts with familiarity such that this transfer feature tends to be important when familiarity is not available. Conversely, the CEFR-J level has the highest Spearman’s $|\rho|$ across all L1s (up to $-.66$ for Chinese) but is only moderately important because it is collinear with other familiarity features, e.g., frequency and EFLLex-level span.

We briefly define features that are not self-explanatory:

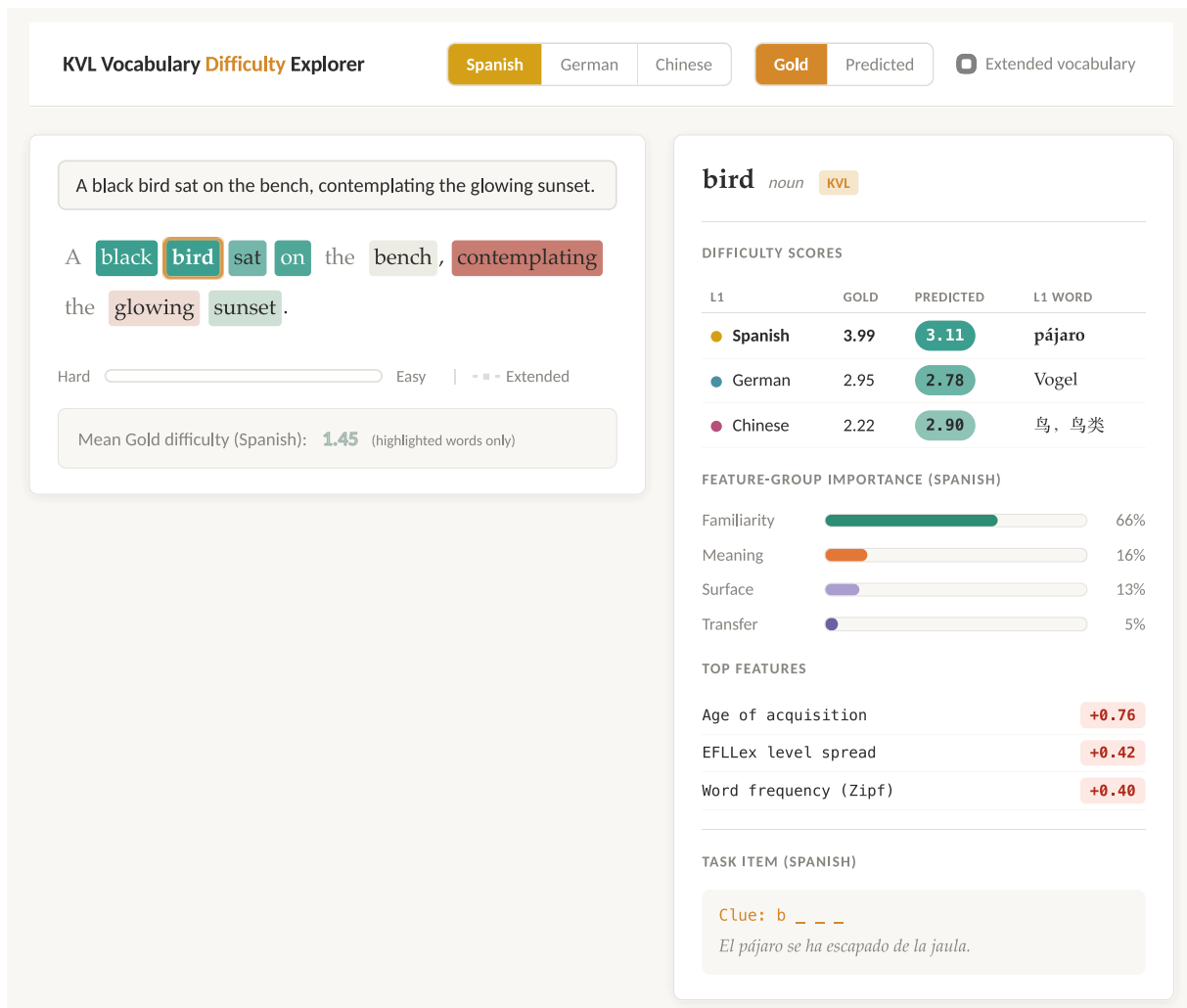


Figure 6: Screenshot of our [interactive demo](#). Words of the input text are highlighted according to their lexical difficulty. Clicking on a word opens a panel that shows the gold-label and predicted difficulty as well as the feature-group importance. The L1-background can be switched.

Logarithmic frequency. Log-transformed word frequency on the Zipf scale (Van Heuven et al., 2014)

$$f_{\text{Zipf}} = \log_{10}(\text{fpmw}) + 3 \quad (1)$$

which modifies the standardized measure of frequency per million words (fpmw) by a logarithmic transformation and adding 3 to keep the measure positive even for low-frequency words, making it more intuitive. We assign a value below the minimal frequency threshold, $f_{\min} = 0.5$, to words missing from the SUBTLEX-UK list.

Contextual diversity. The proportion of film or television programs in SUBTLEX-UK containing at least one occurrence of the word, reflecting the breadth of contexts in which a learner might encounter it (Van Heuven et al., 2014). We expect that words that occur only in certain domains are

less familiar (and thus more difficult to learn) relative to words of the same frequency that occur across many domains. This feature turns out to be especially relevant for Chinese-speaking learners.

EFLLex level span. The number of Common European Framework of Reference for languages (CEFR) levels (Europarat, 2011), A1, A2, B1, B2, C1, at which the word has non-zero frequency in the EFLLex learner corpus (Dürlich and François, 2018). A span of 5 means the word appears at every level from A1 to C1 and is thus broadly familiar, whereas a span of 1 means it is confined to a single proficiency band.

Embedding norm. The ℓ_2 norm of the 300-dimensional English fastText embedding $\mathbf{e}$ (Bojanowski et al., 2017),

$$\|\mathbf{e}\|_2 = \sqrt{\sum_{i=1}^{300} e_i^2}. \quad (2)$$

	Feature	Definition	Source	Mean SHAP			ρ (value-gold)		
				ES	DE	CN	ES	DE	CN
Familiarity	Log frequency	Log-scaled word frequency	SUBTLEX-UK ¹	0.26	0.20	0.19	.40	.43	.60
	Contextual diversity	Distinct film contexts	SUBTLEX-UK ¹	0.05	0.06	0.16	.37	.39	.58
	Age of acquisition	Mean AoA rating	AoA ²	0.27	0.27	0.20	-.44	-.46	-.55
	Percent known	% of raters knowing the word	AoA ²	0.02	0.03	0.02	.17	.18	.22
	CEFR-J level	Vocabulary level (A1 = 1 ... C2 = 6)	CEFR-J ³	0.15	0.17	0.23	-.55	-.57	-.66
	EFLLex-level span	CEFR levels with non-zero frequency	EFLLex ⁴	0.25	0.22	0.31	.43	.45	.64
	EFLLex A1	Learner-corpus token share at A1	EFLLex ⁴	0.06	0.04	0.02	.40	.42	.49
	EFLLex A2	" at A2	"	0.13	0.10	0.05	.40	.48	.52
	EFLLex B1	" at B1	"	0.05	0.06	0.03	.30	.36	.46
	EFLLex B2	" at B2	"	0.03	0.05	0.08	.16	.18	.34
	EFLLex C1	" at C1	"	0.03	0.03	0.03	.05	.05	.18
Meaning	Embedding norm	fastText embedding ℓ_2 norm	fastText ⁵	0.23	0.18	0.08	.32	.33	.33
	Hypernym depth	Mean synset depth in hypernym tree	WordNet ⁶	0.13	0.11	0.03	.15	.18	.05
	Sense count	Number of synsets for word + POS	WordNet ⁶	0.07	0.04	0.03	.12	.14	.24
	POS dominance ratio	Token share of dominant POS	SUBTLEX-UK ¹	0.04	0.03	0.05	.00	-.01	-.07
	L1 confusor flag	L1 source had disambiguation annotation	KVL ⁷ (derived)	0.02	0.05	0.01	-.08	-.12	-.09
Surface	Target word length	Character length of English word	KVL ⁷ (derived)	0.04	0.04	0.10	-.33	-.35	-.43
	Source word length	Character length of L1 lemma	KVL ⁷ (derived)	0.04	0.05	0.15	-.23	-.36	-.33
	Syllable count	Estimated syllables of English word	KVL ⁷ (derived)	0.05	0.05	0.05	-.27	-.32	-.37
	Letters per phoneme	Orthographic transparency proxy	AoA ²	0.04	0.04	0.02	.08	.01	.10
	Context sent. length	Character length of L1 context sentence	KVL ⁷ (derived)	0.10	0.07	0.11	-.21	-.24	-.30
	Clue letter	First letter of the spelling clue	KVL ⁷	0.18	0.12	0.16	—	—	—
	L1 initial letter	First letter of the L1 lemma	KVL ⁷ (derived)	0.06	0.07	0.07	—	—	—
Transfer	Character similarity	Char. n -gram TF-IDF cosine, EN vs. L1	KVL ⁷ (derived)	0.51	0.52	0.00	.10	.25	—

¹Van Heuven et al. (2014); ²Kuperman et al. (2012); ³Negishi et al. (2013); ⁴Dürlich and François (2018);

⁵Bojanowski et al. (2017); ⁶Miller (1995); Fellbaum (1998); ⁷Schmitt et al. (2021); Skidmore et al. (2025)

Table 2: Mean |SHAP| importance per feature and L1, averaged over test-set items across 20 random seeds (5–95% range < 0.02 for all features and L1s), and Spearman’s ρ between feature value and gold lexical difficulty. Features are grouped by category. The five highest |SHAP| values per L1 column are set in **bold**.

Since function words and polysemous words tend to have smaller norms because their embeddings are pulled towards the origin of the space by diverse contexts, concrete and semantically specific words tend to have larger norms (Oyama et al., 2023).

Character similarity. We represent the L1 word and its English translation as character n -gram TF-IDF vectors ($n \in \{2, 3, 4\}$, sublinear TF) and compute their cosine similarity. For Chinese, all values are approximately zero because there is no character overlap between Latin and Chinese script. We give a more detailed definition in Appendix D.

POS dominance ratio. The proportion of SUBTLEX-UK tokens for a lemma that carry its most frequent part of speech (POS), derived from the POS frequency in SUBTLEX-UK (Van Heuven et al., 2014). Values near 1 indicate unambiguous POS. Lower values indicate that the word is frequently used as multiple parts of speech (e.g., light as noun, verb, and adjective).

Letters per phoneme. The ratio of character length to phoneme count, derived from the number of phonemes per word in the Kuperman et al. (2012) dataset. This serves as a proxy for orthographic transparency: Words with many letters per phoneme have less predictable spellings, e.g., thought /θɔ:t/ (7 letters / 3 phonemes = 2.33).

Clue letter and L1 initial letter. These categorical features encode the first letter of the English spelling clue and the first letter of the L1 lemma, respectively. They capture letter-level priming effects: certain initial letters may be associated with easier or harder vocabulary on average, and letter overlap between the clue and the L1 form may facilitate recall. Spearman correlations are not reported for categorical features. Interestingly, the L1-first-letter priming is equally important for Chinese as for Spanish and German (0.06–0.07) as the model learns a baseline for every Chinese word-initial character.

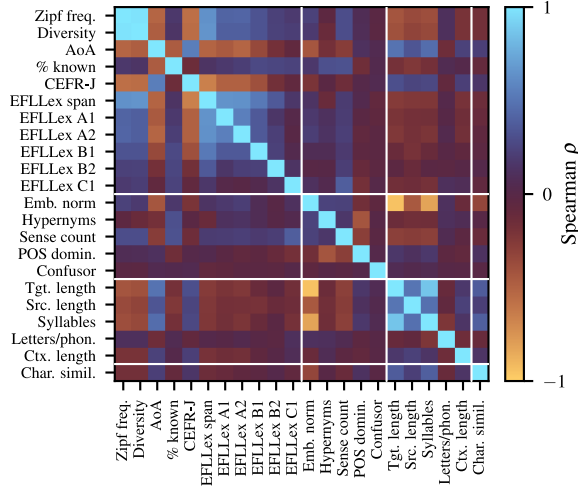


Figure 7: Pairwise Spearman correlation between numeric features (Spanish; the pattern is similar for German and Chinese except for L1-dependent features such as source word length and character similarity). Several pairs are substantially correlated. White lines indicate feature groups.

Figure 7 shows pairwise correlations between the numeric features from Table 2. Some features are strongly correlated, particularly the familiarity features (e.g., logarithmic frequency, contextual diversity, and EFLLex level span) and the surface features (target word length, syllable count, and source word length). This collinearity means that individual SHAP values within a group partly reflect shared variance, which motivates our decision to group the features for the main analysis.

Additional features were evaluated during development but excluded from the final model after unstructured ablation showed no consistent improvement. These include: character-level surprisal, normalized Levenshtein edit distance, longest common subsequence ratios, EFLLex entropy and mean level, WordNet synonym count and derivational family size, POS competition flag, and Spanish-specific CORPES frequencies and anglicism indicators. We provide these in Table 3 for completeness. These features turn out to be too redundant to yield higher performance. For example, edit distance captures the same signal as character similarity but is less informative in this context. We give an exemplary analysis in Appendix H.

D Character similarity

To approximate cross-linguistic form similarity between an English word and its L1 translation, we compare their character- n -gram overlap. This fea-

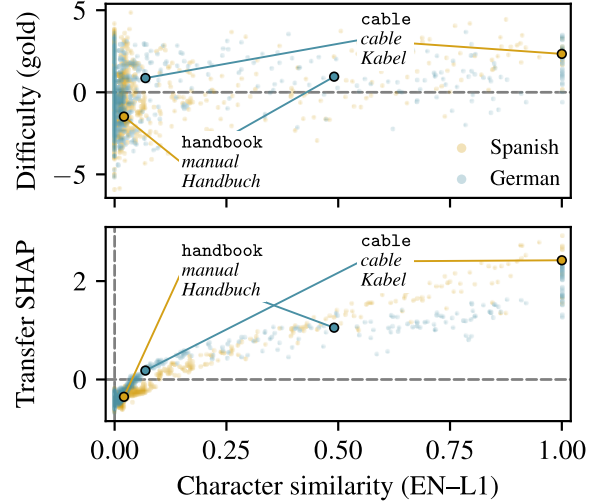


Figure 8: Character similarity between English words and their L1 translations (Spanish, German) versus gold-label difficulty (top) and Shapley-value contribution of this feature (bottom). Two words are highlighted: cable and handbook.

ture is motivated by the intuition that words sharing many letter sequences (e.g., *fantasy* and *fantasía*) can be readily recognized as cognates.

From binary overlap to weighted similarity. In a naive approach, we could represent each word as a binary vector over character n -grams, where each entry states whether a given n -gram occurs in the word, and then compute the cosine similarity between two vectors. This approach would treat all n -grams the same, yet some n -grams might be more informative than others.

To account for the informativeness of an n -gram, we weight it using *term-frequency-inverse-document-frequency* (TF-IDF). In this context, words correspond to “documents” and n -grams with $n \in \{2, 3, 4\}$ to “terms”. The TF-IDF weight of an n -gram t in a word w is implemented as

$$\text{tf-idf}(t, w) = (1 + \log(\text{tf}(t, w))) \log\left(\frac{N}{\text{df}(t)}\right), \quad (3)$$

where $\text{tf}(t, w)$ is the frequency of t in w , $\text{df}(t)$ is the number of words in the training data containing t , and N is the total number of word forms in the union of English and L1 word forms in the training data. We use sublinear term-frequency scaling, i.e., $1 + \log(\text{tf}(t, w))$ instead of $\text{tf}(t, w)$, so that repeated occurrences are weighted less than linearly.

Cosine similarity. We can thus represent each word as a TF-IDF-weighted vector over character

Feature	Definition
Character surprisal	Mean $-\log(p(\text{char}))$ under training-set unigram distribution
Edit distance (norm.)	Levenshtein distance between EN and L1 word, normalized by $\sqrt{\text{len}_{\text{EN}}} \cdot \sqrt{\text{len}_{\text{L1}}}$
LCS ratio (EN)	Longest common subsequence length / target word length
LCS ratio (L1)	Longest common subsequence length / source word length
EFLLex entropy	Normalized Shannon entropy of the CEFR-level frequency distribution
EFLLex mean level	Frequency-weighted mean CEFR level
WN synonym count	Number of unique lemma names across all synsets
WN derivational family	Number of derivationally related forms in WordNet
WN POS count	Number of distinct parts of speech in WordNet
POS competition	1 if the tested POS differs from the dominant SUBTLEX-UK POS
Exact match	1 if EN word = L1 lemma after normalization
L1 second letter	Second letter of the L1 lemma
CORPES frequency (ES)	Log-transformed normalized frequency in the Spanish CORPES corpus
CORPES dispersion (ES)	Log-transformed dispersion across CORPES subcorpora
Anglicism frequency (ES)	CORPES frequency of the English word looked up in Spanish text
Anglicism flag (ES)	1 if the L1 lemma appears in a Wiktionary-derived anglicism list

Table 3: Additional features evaluated on the development set but excluded from the final model after ablation showed no consistent improvement of model performance. CORPES refers to [Real Academia Española \(2025\)](#).

n -grams. Cross-linguistic form similarity is then measured as the cosine similarity between the English word w_{en} and its L1 translation w_{L1} :

$$\text{sim}(w_{\text{en}}, w_{\text{L1}}) = \frac{\mathbf{x}_{w_{\text{en}}} \cdot \mathbf{x}_{w_{\text{L1}}}}{\|\mathbf{x}_{w_{\text{en}}}\| \|\mathbf{x}_{w_{\text{L1}}}\|}. \quad (4)$$

The TF-IDF vectors are ℓ_2 -normalized, so cosine similarity corresponds to their dot product.

Empirical behavior. Figure 8 illustrates how this character similarity relates to lexical difficulty and to its feature importance in Spanish- and German-L1 model predictions. Most items have near-zero similarity, with some items spread across the range up to perfect matches at 1. Higher similarity is associated with lower difficulty, cf. Table 2. The Shapley values show that high similarity contributes positively to predictions above a small threshold at approximately 0.05–0.10. Below this threshold, negligible similarity is a cue for higher difficulty. This nonlinear distribution explains why character similarity has high average feature importance despite only a moderate correlation with difficulty.

E Hyperparameters

Relevant hyperparameters are summarized in Table 4. The hyperparameters were tuned on the Spanish development set. Reported results are based on 20 random seeds.

F Correlation of difficulties across L1s

Figure 9 shows pairwise correlations of gold-label difficulty across L1s. All language pairs are positively correlated, quantifying the shared component

Component	Parameter	Value
CatBoost	Loss function	RMSE
	Tree depth	7
	Learning rate	0.017
	Iterations	2,400
	ℓ_2 leaf regularization	0.8
Evaluation	Eval. seeds	20 (1, 8, 15, ..., 134)
	Submission seeds	3 (42, 142, 242)
	Metrics	RMSE (primary) Pearson’s r
Baseline	Model	Ridge regression
	ℓ_2 regularization	1

Table 4: CatBoost model configuration, evaluation setup, and baseline.

of lexical difficulty: Words that are hard for one L1 group tend to be hard for others. However, the correlations are far from perfect, indicating that L1-specific factors account for a substantial portion of the variance.

G Relationship of frequency and transfer

In Fig. 10, panel (a) shows lexical difficulty in the plane of character similarity and word frequency. We observe that the easiest words are frequent or cognate. Those words that are both cognate and frequent (items in the top right) shape the model predictions equally through both features, as the Shapley values in panels (b) and (c) indicate. The thresholds for the Shapley values that the model learns appear largely independent for these two features: a vertical divide.

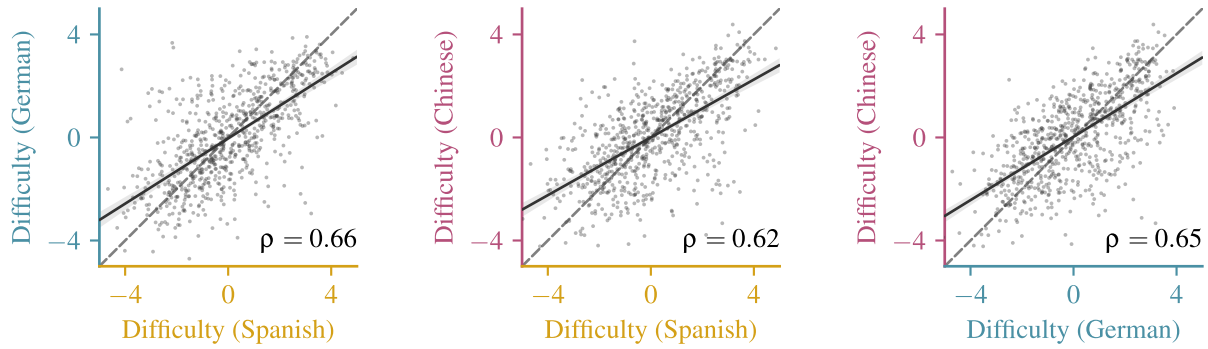


Figure 9: Pairwise correlation of gold-label lexical difficulty across L1s. Each point is one English word tested in both languages. Pearson’s r is shown per pair. The shared variance reflects universal difficulty factors and the unexplained variance motivates L1-specific modeling.

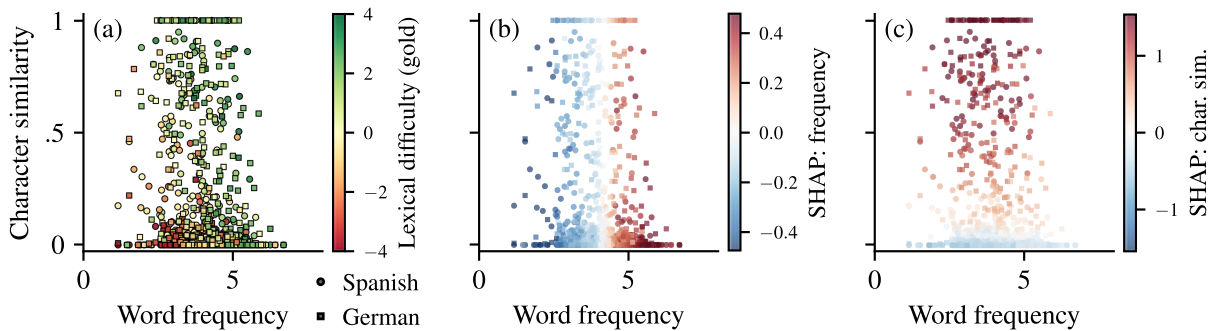


Figure 10: Character similarity by word frequency for Spanish and German, colored by (a) lexical difficulty and the feature importances of (b) word frequency and (c) character similarity.

H Feature analysis

In this section, we examine several features that we evaluated during development but excluded from the final model, cf. Table 3. The goal is to show how these features are related to lexical difficulty and why they may be redundant.

POS competition. Some English word forms occur frequently with one part of speech while another is less common, e.g., received as an adjective meaning “accepted” or bar as verb rather than a noun. We define a binary POS-competition flag, i.e., whether the POS of the target word differs from the most frequent POS.

We test whether POS competition affects the prediction error by comparing the prediction errors (gold – predicted) for items with and without POS competition, see Fig. 11. We find, across all L1s, no significant difference in mean or central tendency (e.g., Spanish: Welch’s t -test: $p = 0.09$, Wilcoxon–Mann–Whitney test: $p = 0.36$), indicating that POS competition does not introduce a systematic model-prediction bias. Furthermore, items with POS competition do not differ signifi-

cantly in gold-label difficulty from items without competition (Welch’s t -tests and Wilcoxon–Mann–Whitney tests, all $p > 0.20$).

However, POS competition has a robust effect on the dispersion of the prediction error. Items with POS competition exhibit substantially higher variance across all L1s, with standard deviations increasing by 24–32%, see Table 5. This difference is statistically significant according to both Brown–Forsythe and Fligner–Killeen tests ($p < 0.01$), which suggests that POS competition is more difficult to predict for the model, but does not introduce a systematic bias.

L1	std _{no}	std _{yes}	ratio	BF p	FK p
Spanish	1.16	1.45	1.24	< 0.01	< 0.01
German	1.04	1.37	1.32	< 0.01	< 0.01
Chinese	0.97	1.24	1.28	< 0.01	< 0.01

Table 5: Standard deviation of the prediction error for items with and without POS competition across L1s, with their ratio, and p -values from Brown–Forsythe (BF) and Fligner–Killeen (FK) tests.

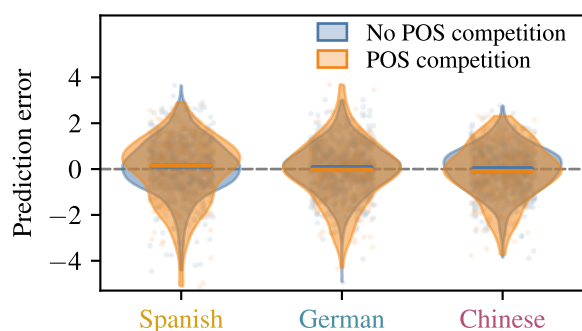


Figure 11: Prediction-error distribution (gold – predicted) by POS competition status across L1 groups (Spanish, German, Chinese). The violin plots compare items without POS competition ($n_{\text{no}} = 585$) versus items with POS competition ($n_{\text{yes}} = 163$) per language, with horizontal lines marking the median.

Edit distance. In addition to character-level cosine similarity, we evaluated character edit distance as a measure of orthographic similarity. While edit distance shows a weak-to-moderate negative correlation with gold-label difficulty for Spanish ($r = -0.17$) and German ($r = -0.30$), it is strongly negatively correlated with character-level cosine similarity for Spanish ($r = -0.77$) and German ($r = -0.73$), see Fig. 12. These two features encode largely the same information, making the inclusion of both redundant. Compared with cosine similarity, edit distance spreads out dissimilar word pairs over a wider range while compressing highly similar forms, making it less informative for distinguishing degrees of cognateness.

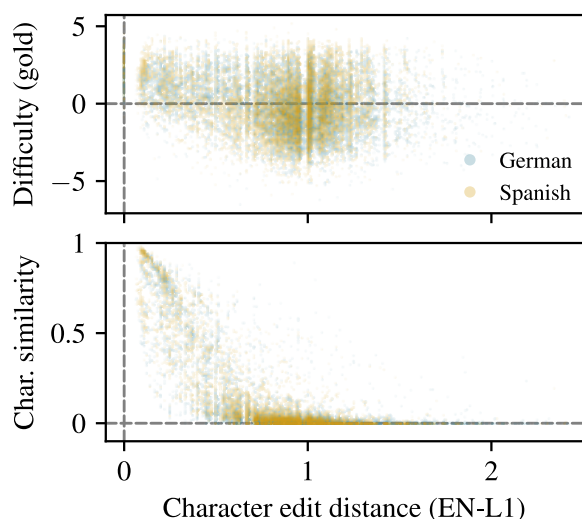


Figure 12: Normalized edit distance between English words and their L1 translations in Spanish and German versus gold-label difficulty (top) and its relationship with character-level cosine similarity (bottom).

L1 frequency. Finally, we examine L1 word frequency, which has also been suggested as a predictor of L2 lexical difficulty, under the assumption that more frequent words in a learner’s native language may facilitate recognition or acquisition (Ellis, 2002; Peters, 2019). Using Spanish as an example, Fig. 13 shows that, although higher L1 frequency is associated with lower difficulty ($r = 0.15$), L1 and L2 frequency are themselves strongly positively correlated ($r = 0.44$). As a result, L2 frequency is largely redundant when L1 frequency is already included.

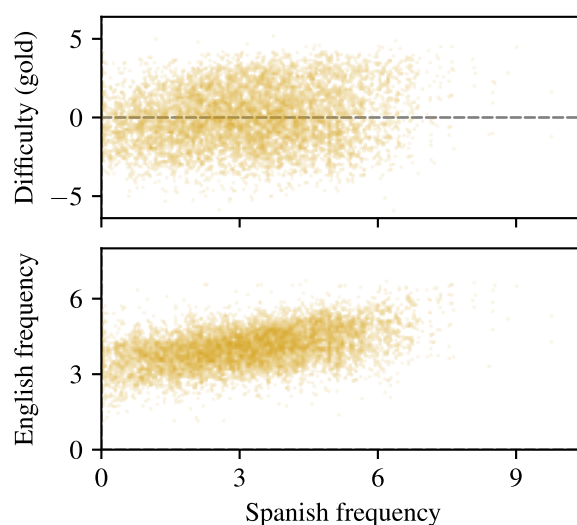


Figure 13: Frequency of L1 (Spanish) source words versus gold-label difficulty (top) and L2 (English) word frequency (bottom).

Submission at BEA 2026 Shared Task 1: Ensemble of Multilingual Encoders with NMT Augmentation for L1-Aware Vocabulary Difficulty Prediction

Bernardo Stearns¹ Thomas Gaillat²
John P. McCrae¹ Jefkine Kafunah¹

¹ Research Ireland Insight Centre for Data Analytics,
Data Science Institute, University of Galway, Ireland

² LIDILE / Université Rennes 2, 35000 Rennes, France

Contact: bstearnsreisendepinho@universityofgalway.ie

Abstract

This paper describes a system submission to the closed track of the BEA 2026 shared task on L1-aware vocabulary difficulty prediction (Spanish, German, Mandarin Chinese). We explored three approaches: hand-crafted tabular features with tree-based regressors, fine-tuned multilingual encoders, and fine-tuned decoders (artificial learner simulation with LoRA-tuned Pythia models). Each was examined with and without an English context produced by neural machine translation (NMT), and we also combined all three families in a single ensemble. In the end, ensembling *only* the fine-tuned multilingual encoders (four base architectures together with four NMT-augmented variants, combined through per-language stacking) gave the best results while remaining the simplest configuration to train.

1 Introduction

Vocabulary knowledge is fundamental to language proficiency, directly influencing what learners can comprehend and produce (Laufer, 1997). Establishing word difficulty is essential for creating level-appropriate educational content and valid assessment instruments, yet traditional calibration methods rely on costly expert judgment and pretesting, limiting scalability.

The BEA 2026 shared task (British Council, 2026) addresses this challenge by framing vocabulary difficulty prediction as a regression problem: given an English word and its associated context, predict its difficulty for learners with a specific first language (L1) background. The task uses the British Council’s Knowledge-based Vocabulary Lists (KVL; Schmitt et al., 2021), a dataset of 6,768 items per L1 with psychometrically calibrated GLMM difficulty scores across three L1 groups: Spanish (Es), German (De), and Mandarin Chinese (Cn). Two tracks are offered: *closed* (sep-

arate per-language models) and *open* (a single multilingual model).

While previous shared tasks have addressed related problems, such as complex word identification (Paetzold and Specia, 2016; Yimam et al., 2018) and lexical complexity prediction (Shardlow et al., 2021, 2024), they did not consider the learner’s L1 background, which is known to strongly influence L2 vocabulary acquisition through crosslinguistic transfer (Jarvis and Pavlenko, 2008). The last BEA language learning shared task was on grammatical error correction in 2019 (Bryant et al., 2019).

Our submission makes the following contributions:

1. A systematic comparison of 11 multilingual encoder architectures for L1-aware difficulty prediction;
2. Encoder training augmented with neural machine translation (NMT) using three translation systems with two decoding modes, providing complementary crosslinguistic signal;
3. An ensemble of eight encoders with per-language optimized stacking weights;
4. A tabular feature study combining multilingual embeddings, typological, phonetic, CEFR/WordNet, and cognate features with XGBoost and LightGBM regressors;
5. A scaling study of decoder-based artificial learner simulation (ALS) using Pythia models from 70M to 1.4B parameters.

2 Related Work

Lexical Complexity and Difficulty. The SemEval-2016 CWI shared task (Paetzold and Specia, 2016) introduced binary complex word identification, extended to multilingual settings by

Yimam et al. (2018). Shardlow et al. (2021) moved to continuous complexity prediction with the LCP shared task, and Shardlow et al. (2024) addressed multilingual lexical simplification. However, none of these tasks incorporated learner L1 as a predictive factor, even though there is longstanding evidence that L1 background shapes L2 vocabulary processing and difficulty.

L1 Influence on L2 Vocabulary. Crosslinguistic influence research has established that a learner’s L1 systematically affects L2 vocabulary acquisition (Jarvis and Pavlenko, 2008). Cognates are easier to learn, while false friends create difficulty. This motivates L1-aware prediction models that can capture these transfer effects. This L1 influence has also been examined within language model-based approaches that simulate L2 learners, offering a complementary computational lens on the same phenomenon.

Artificial Learner Simulation. Several lines of work adapt or probe language models to simulate L2 processing and production.

Aoyama and Schneider (2024) pretrain GPT-2 sequentially on six different L1s and then on English as L2. The resulting “L2LMs” produce word surprisal that predicts nonnative reading times, with effects most pronounced for L1 backgrounds typologically distant from English.

In a related direction, Stearns et al. (2024) pretrain BERT on EFCAMDAT – the Education First-Cambridge Open Language Database (Geertzen et al., 2013), a publicly released corpus of ~83M words of English-as-a-foreign-language essays from ~550k EF Englishtown learners with self-declared L1 and CEFR-aligned proficiency labels – and C4200M – a 200M-sentence synthetic learner-error corpus built by injecting grammatical errors into clean C4 web text via a tagged corruption model, approximating the distribution of L2 mistakes at web scale (Stahlberg and Kumar, 2021) – and probe the resulting “artificial learners” via masked-token prediction, finding systematic divergence from a generic native model that is largest at lower CEFR levels.

Hu and Cong (2025) apply LLM-derived surprisal as a proficiency signal for Chinese L2 writing. They compare a multilingual, a Chinese-general, and a Traditional-Chinese-specific LLM on TOCFL essays across CEFR levels A2–C1, and show that surprisal differentiates proficiency and correlates with classic lexical and syntactic com-

plexity indices – evidence that LLM-based probes apply cross-linguistically beyond English.

We extend this artificial-learner line to vocabulary difficulty regression by using Pythia decoder models (Biderman et al., 2023) with LoRA adaptation (Hu et al., 2022) as a next-token-prediction proxy for item difficulty.

3 System Description

3.1 Task and Data

The KVL dataset provides 6,768 parallel vocabulary items across three L1 groups (Spanish, German, Mandarin Chinese), split into training (~6,091 items/language) and development (677 items/language) sets. Each item consists of:

- en_word: the target English word
- en_clue: a letter clue for the word
- L1_word: the L1 translation
- L1_context: an L1 contextual prompt
- GLMM_score: the target difficulty (lower = harder)

Items with the same item_id across L1 files refer to the same English word, enabling cross-language analysis.

3.2 Encoder Models

We fine-tune 11 multilingual encoder architectures for regression, using the full model parameters (no adapter layers). The input is constructed by concatenating all available text fields separated by [SEP] tokens in the order: L1_word, L1_context, en_word, en_clue. A single regression head maps the [CLS] representation to the predicted difficulty score.

Table 1 summarizes the training configuration. All models use identical hyperparameters: learning rate 1×10^{-5} , batch size 8 (effective 16 with gradient accumulation), 10 epochs with early stopping (patience 3), AdamW optimizer with weight decay 0.01 and 10% warmup, and FP16 mixed precision. The maximum sequence length is 256 tokens. We select the best checkpoint by development-set Pearson correlation. The regression target is the GLMM lexical-difficulty score provided by the organizers; the head is trained with mean-squared-error loss.

The 11 architectures tested are: Multilingual E5-Large (Wang et al., 2022), XLM-R Large and Base (Conneau et al., 2020), RemBERT (Chung

Hyperparameter	Value
Learning rate	1×10^{-5}
Batch size (effective)	16
Max epochs	10
Early stopping patience	3
Max sequence length	256
Classifier dropout	0.1
Optimizer	AdamW
Weight decay	0.01
Warmup ratio	0.10
Precision	FP16

Table 1: Encoder training hyperparameters.

et al., 2021), InfoXLM-Large (Chi et al., 2021a), Paraphrase Multilingual MPNet (Reimers and Gurevych, 2019, 2020), Multilingual E5-Base, mBERT (Devlin et al., 2019), CANINE-S (Clark et al., 2022), XLM-Align Base (Chi et al., 2021b), and Chinese RoBERTa Large (Liu et al., 2019). All checkpoints were obtained from the HuggingFace Hub. Of these, the top four (E5-Large, XLM-R Large, RemBERT, InfoXLM-Large) were selected for the ensemble and NMT augmentation experiments.

3.3 NMT Augmentation

We hypothesize that augmenting the L1 context with English translations from NMT systems provides complementary crosslinguistic signal, effectively giving the encoder both the L1 perspective and multiple L2 (English) interpretations of the same content. We generate translations using three NMT systems of varying capacity:

- **OPUS-MT** (Tiedemann and Thottingal, 2020): ~ 77 M parameters
- **NLLB-600M**: distilled 600M model (NLLB Team et al., 2022)
- **NLLB-1.3B**: full 1.3B model (NLLB Team et al., 2022)

For each system, we produce translations in two decoding modes: (1) *free beam search*, where the model generates its own English translation from the L1 input, and (2) *forced decoding*, where the model is constrained to produce the target English word. This yields six additional translated text strings per item.

The NMT-augmented encoders use the same four top architectures but with an extended input sequence that interleaves the original L1 context with these translated L2 contexts: L1_word, L1_context, followed by the six NMT translations,

Spanish item, target “flock”	
L1_word	rebaño
L1_context	Un ___ de ovejas pasta en el campo.
OPUS free	A herd of sheep grazes in the field.
OPUS forced	A flock of sheep grazes in the field.
NLLB-600M free	A flock of sheep grazes in the field.
NLLB-600M forced	A flock of sheep grazes in the field.
NLLB-1.3B free	A flock of sheep is grazing in the field.
NLLB-1.3B forced	A flock of sheep grazes in the field.

Table 2: Worked example of NMT-augmented input for a Spanish item. Three NMT systems each emit a free and a forced decode; the six resulting English strings are concatenated to the L1 fields. Disagreements between free and forced decodes (OPUS’s “herd” vs. the gold “flock”) expose the encoder to lexical alternatives consistent with the L1 context.

then en_word, en_clue, all concatenated as text with [SEP] separators. The maximum sequence length is increased to 384 tokens to accommodate the longer input. Batch size is increased to 12; all other hyperparameters remain identical.

3.4 Artificial Learner Simulation

We hypothesize that diversifying the model’s embedding space away from native-speaker distributional norms and towards representations conditioned on learner production (which we refer to as *learner embeddings*) provides complementary predictive signal for vocabulary difficulty: words that are hard for L2 learners may be unremarkable under a native prior but surface as atypical once the model’s lexical distribution has been shifted towards L2 usage. Following this intuition, our *artificial learner simulation* (ALS) pipeline shifts a Pythia (Biderman et al., 2023) decoder’s next-token distribution towards L2 learner production and uses the resulting learner-conditioned representation as the input to the regression head. We chose decoder-based rather than encoder-based artificial learners to maximise architectural diversity relative to the multilingual encoders of Section 3.2; encoder artificial learners remain a separate ablation we did not run.

Concretely, the ALS pipeline has two stages: (i) a learner-style language-modelling pretraining on EFCAMDAT that yields an *artificial learner*

checkpoint, and (ii) a regression adaptation of that checkpoint to the shared-task data.

Stage 1: EFCAMDAT pretraining. We continue the causal-LM pretraining of Pythia (Biderman et al., 2023) at five scales (70M, 160M, 410M, 1B, 1.4B) on the full EFCAMDAT corpus (Geertzen et al., 2013) of L2 English learner essays, using the standard next-token prediction objective and no modification to the model architecture. After this stage, each checkpoint (pythia-{size}-all-data) is a decoder whose distribution over continuations reflects the lexical and grammatical choices of L2 learners rather than those of a native reference corpus.

Stage 2: Regression adaptation. To transfer an artificial-learner checkpoint to the shared task, we discard the pretrained LM head and attach a randomly initialised 2-layer MLP regression head of the form Dropout $\rightarrow$ Linear($h, h/2$) $\rightarrow$ GELU $\rightarrow$ Dropout $\rightarrow$ Linear($h/2, 1$), where h is the backbone hidden size. Items are serialised as en_word | L1_word | en_clue | L1_context followed by the suffix “Difficulty:”, left-padded, and the sequence representation is obtained by taking the hidden state of the last non-padding token, the decoder-appropriate analogue of the [CLS] pooling used by the encoder models of Section 3.2. The backbone itself is frozen except for LoRA adapters (Hu et al., 2022) with rank $r = 16$, $\alpha = 32$, dropout 0.05, applied to the query_key_value and dense projections of every GPT-NeoX block; the regression head is trained at full precision. Training minimises MSE with AdamW, learning rate 2×10^{-5} , effective batch size 16, up to 10 epochs with early stopping on dev Pearson r .

3.5 Tabular Features

We additionally extract a set of hand-crafted features organized into nine modules. For each module below we report the implementation actually used:

- **Embeddings** (1,154): Multilingual E5-Large sentence embeddings for the L1 context, target word, and source, plus pairwise cosine similarities; computed with sentence-transformers (Reimers and Gurevych, 2019).
- **Language pair typology** (226): morphological and typological features comparing

each L1 to English, looked up from a hyper-granular morphological-features table derived from WALS (Dryer and Haspelmath, 2013) and expressed as both raw L1 values and L1–English difference vectors.

- **IPA phonetic** (95): articulatory, sonority, and syllable-structure features over the G2P-converted target word, computed with panphon’s FeatureTable and Distance modules (Mortensen et al., 2016).
- **CEFR + WordNet** (12): per-word CEFR statistics (presence, sense count, min/max/mean/range level, and binary indicators for A1–C2), obtained from a sensekey-to-CEFR lookup table built over WordNet (Fellbaum, 1998) senses.
- **Cognate similarity**: Levenshtein distance, longest-common-subsequence length, character-bigram Jaccard, and shared-prefix length between the English target and its L1 translation; all implemented as plain Python, no external distance library.
- **Clue features**: length, revealed-positions length, vowel presence, and blank-to-letter ratio over the en_clue field.
- **Counts / POS / L1**: string-length and word-count statistics; one-hot encoding of a pre-annotated en_target_pos column; one-hot L1 language indicators (es/de/cn).
- **NMT-derived**: five features per NMT system capturing whether the target word appears in the free translation, the normalised Levenshtein distance between forced and free decodes, the Pythia perplexity of the forced translation, the forced/free perplexity ratio, and the NMT-score drop.
- **Frequency**: Zipf-scaled word-frequency bands for the English target, retrieved via wordfreq (Speer, 2022).

We fit regressors from the scikit-learn toolkit (Pedregosa et al., 2011), covering linear regression, Ridge, Lasso, decision trees, and tree ensembles (random forest, extra trees, gradient boosting, AdaBoost), together with XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017). The best-performing tabular configuration combines all feature groups with ALS-derived

embeddings (all_concat_als_xgb, $r = 0.797$), using XGBRegressor with $n_{\text{estimators}} = 1000$, $\text{max_depth} = 6$, learning rate 0.03, and L_1/L_2 regularisation coefficients 0.1. Feature extraction relied on sentence-transformers (Reimers and Gurevych, 2019) for the 1,154 Multilingual E5-Large embeddings and panphon (Mortensen et al., 2016) for the 95 articulatory, sonority, and syllable-structure features. The full per-configuration breakdown across ~ 15 feature/regressor combinations is reported in Appendix A.

3.6 Ensemble Strategy

Our final system ensembles the predictions of eight encoder models: four base and four NMT-augmented (Section 3.3). We experimented with two stacking strategies:

Nelder-Mead. We compute a weighted average of predictions: $\hat{y} = \sum_{i=1}^8 w_i \cdot \hat{y}_i$, where weights $w_i \geq 0$ and $\sum w_i = 1$. The weights are fit *separately per language* using the Nelder-Mead simplex method (Nelder and Mead, 1965), as implemented in `scipy.optimize.minimize` (Virtanen et al., 2020) with $\text{xatol} = 10^{-6}$ and $\text{fatol} = 10^{-8}$, to maximize Pearson correlation on the 677 development items. This is a non-learning approach: no meta-model is trained, and the only free parameters are the 8 weights per language.

ElasticNet Meta-Learner. We train a scikit-learn (Pedregosa et al., 2011) ElasticNet regressor (Zou and Hastie, 2005) ($\alpha = 0.1$, $l_1\text{-ratio} = 0.5$) on the training-set predictions of the eight encoders.

4 Results

4.1 Individual Encoder Performance

Table 3 presents the development-set results for all 11 encoder architectures. Multilingual E5-Large achieves the best individual performance ($r = 0.845$), followed closely by XLM-R Large ($r = 0.843$) and RemBERT ($r = 0.843$). There is a clear divide between the top-four large models and the rest: the gap between 4th (InfoXLM-Large, $r = 0.829$) and 5th (Paraphrase MPNet, $r = 0.810$) is substantial.

4.2 NMT Augmentation Effect

Table 3 shows the impact of NMT augmentation on the top-four encoders. For each architecture, we compare the base model with its NMT-augmented variant (using all six NMT contexts:

OPUS-MT, NLLB-600M, NLLB-1.3B, each with free and forced decoding). NMT augmentation consistently improves Pearson r across all four architectures, with the largest gain for InfoXLM-Large (+0.021). RMSE improves for XLM-R Large and RemBERT but slightly worsens for E5-Large and InfoXLM-Large.

4.3 Artificial Learner Simulation

Table 3 presents the ALS scaling study. Performance scales monotonically with model size, from $r = 0.494$ (70M) to $r = 0.812$ (1.4B). However, even the largest ALS model underperforms the best individual encoder ($r = 0.845$), suggesting that for this task, direct fine-tuning of encoder representations is more effective than the decoder-based ALS approach. Several follow-up experiments remain open, including the NMT-augmented variant of the 1.4B checkpoint, larger Pythia scales, encoder-based artificial learners, and adding the Pythia ALS models to the final ensemble alongside the multilingual encoders; we could not run them within the shared-task timeline and leave them to future work.

4.4 Tabular Feature Models

We trained approximately 40 scikit-learn regressors over the hand-crafted feature modules described in Section 3.5, formed by combining feature subsets with five regressor families (linear regression, ridge, decision tree, XGBoost, LightGBM). The strongest configuration, all_concat_als_xgb, concatenates all features with ALS-derived embeddings and is fit with XGBoost, reaching $r = 0.797$ on the development set. This result surpasses three of the base encoders (XLM-R Base, $r = 0.791$; mBERT, $r = 0.761$; CANINE-S, $r = 0.668$) and is comparable to Pythia 1B ALS ($r = 0.793$), but remains below the seven stronger base encoders and all NMT-augmented encoder variants. We retain the top two tree-based configurations for the 10-model ensemble experiments, but observe no improvement in results. The full per-configuration results appear in Appendix A.

4.5 Ensemble Results (Development Set)

Table 3 presents our ensemble results on the development set. The 8-encoder XGBoost blend (a learned non-linear meta-model) achieves $r = 0.882$, a +3.7% absolute improvement over the best single encoder. No-learning strategies (simple average, Nelder-Mead, hill climbing) cluster tightly between $r = 0.870$ and $r = 0.871$, and the ElasticNet meta-learner achieves $r = 0.871$. On hidden

System	Es	De	Cn	Avg r	RMSE↓ (primary)
<i>Individual encoders — base (dev)</i>					
E5-Large	.843	.843	.851	.845	1.025
XLM-R Large (v1)	.840	.845	.845	.843	1.085
RemBERT	.840	.840	.848	.843	0.980
XLM-R Large (v2)	.833	.843	.843	.839	1.105
InfoXLM-Large	.827	.826	.833	.829	1.166
Para. MPNet	.808	.806	.816	.810	1.094
E5-Base	.804	.803	.812	.806	1.166
XLM-R Base	.789	.787	.796	.791	1.195
mBERT	.758	.755	.768	.761	1.202
CANINE-S	.665	.668	.672	.668	1.364
<i>Individual encoders — NMT-augmented (dev)</i>					
E5-Large +NMT	.852	.857	.841	.850	1.046
XLM-R Large +NMT	.855	.854	.835	.849	0.994
RemBERT +NMT	.854	.845	.842	.848	0.963
InfoXLM-Large +NMT	.849	.858	.842	.850	1.043
<i>Individual decoders — artificial learner simulation, no NMT (dev, seed 42)</i>					
Pythia 70M	.481	.530	.465	.494	1.512
Pythia 160M	.666	.601	.590	.622	1.362
Pythia 1B	.801	.790	.785	.793	1.068
Pythia 1.4B	.824	.822	.790	.812	1.030
<i>Individual decoders — artificial learner simulation, NMT features (dev, seed 42)</i>					
Pythia 70M +NMT (single, opus-mt)	.580	.561	.539	.561	1.440
Pythia 70M +NMT (all 3 systems)	.563	.557	.549	.557	1.444
<i>Tabular / scikit-learn models (dev; best of ~15, full breakdown in Appendix A)</i>					
all_concat_als_xgb (best)	.813	.784	.793	.797	1.099
<i>Submitted 8-encoder ensembles (dev; full ablation in Appendix C)</i>					
Nelder-Mead	.875	.870	.868	.871	0.902
ElasticNet meta	.875	.870	.867	.871	0.891
<i>Baselines (shared task)</i>					
Closed	.748	.753	.736	.745	1.287
Open	.787	.800	.804	.797	1.125

Table 3: Train-dev split results across all model families: base and NMT-augmented multilingual encoders, decoder-based artificial learner simulation (Pythia, with and without NMT features), tabular/scikit-learn regressors, the two submitted 8-encoder ensemble strategies (Nelder-Mead and ElasticNet meta-learner), and the shared-task baselines. Per-language Pearson r on the 677-item dev set is shown where reported per language; otherwise overall (pooled) r /RMSE is given. Baseline dev-set RMSE is the average across the three L1s, computed from the organizer-released per-L1 baseline predictions: Closed 1.287 (Es 1.357, De 1.328, Cn 1.175); Open 1.125 (Es 1.206, De 1.149, Cn 1.021). Test-set RMSE for both baselines appears in Table 4. Tabular section shows only the best configuration; the full breakdown is in Appendix A. GPT-2 decoder baselines are reported in Appendix B. Full ensemble-strategy ablation (alternative weighting schemes and 10-model variants) is in Appendix C.

test data, however, the Blend XGB advantage over these simpler strategies disappears (see §5): we therefore selected Nelder-Mead and ElasticNet for official submission, and treat the +0.011 dev-set gap as meta-model overfitting rather than generalisable signal.

Notably, including tabular models (XGBoost, LightGBM) in a 10-model ensemble slightly degraded performance ($r = 0.872$) compared to the 8-encoder-only ensemble, suggesting the tabular models introduced noise rather than complementary signal.

4.6 Official Test Results

Table 4 presents the official test set results for the closed track. Ranked by RMSE (the official metric), our best submission per language reaches 0.975 (Es, Nelder-Mead), 0.903 (De, ElasticNet) and 0.820 (Cn, ElasticNet), placing uogal **2nd** on Spanish and German behind Glite, and **3rd** on Chinese behind Glite and Sakura (which submitted 0.816). All four uogal submissions improve over the closed-track baseline by 0.28–0.36 RMSE points, and even the worse of the two beats the open-track baseline on every language. The same picture in Pearson r – 0.859 (Es), 0.869 (De), 0.880 (Cn) – shows ElasticNet ahead on Spanish/German and Nelder-Mead nominally ahead on Chinese by 0.001, illus-

Lang.	System	r	RMSE ↓ (primary)
Es	Glite (1st)	.877	0.903
	uogal Nelder-Mead	.858	0.975
	uogal ElasticNet	.859	0.977
	Closed Baseline	.765	1.257
	Open Baseline	.783	1.198
De	Glite (1st)	.871	0.885
	uogal ElasticNet	.869	0.903
	uogal Nelder-Mead	.865	0.944
	Closed Baseline	.773	1.258
	Open Baseline	.786	1.166
Cn	Glite (1st)	.889	0.776
	Sakura (2nd)	.874	0.816
	uogal ElasticNet	.879	0.820
	uogal Nelder-Mead	.880	0.841
	Closed Baseline	.753	1.140
	Open Baseline	.804	1.034

Table 4: Official test set results (closed track). uogal ranks 2nd across all three languages, behind Glite.

trating that the two metrics disagree slightly on the Chinese ranking; we treat RMSE as authoritative since it is the official ranking metric.

5 Analysis

Primary Ranking Metric (RMSE). RMSE is the official ranking metric. Our best per-language submission reaches RMSE 0.975 (Es), 0.903 (De) and 0.820 (Cn), beating the closed-track baseline by 0.28–0.36 points (Table 4). On Chinese the two metrics order our submissions differently (r marginally prefers Nelder-Mead, RMSE clearly prefers ElasticNet); we follow the RMSE verdict throughout.

Superiority of Ensembling. Ensembling provides a substantial gain over the best individual model on the official metric. The best base encoder reaches RMSE 0.980 (RemBERT) and the best NMT-augmented encoder reaches 0.963 (RemBERT +NMT). On the dev set the 8-encoder Blend XGB meta-model reaches RMSE 0.851, a 0.13 absolute improvement over the best base encoder and 0.11 over the best NMT-augmented one (Table 3); the no-learning and linear-meta strategies (simple averaging, weighted-RMSE, weighted-Pearson, hill climbing, Nelder-Mead, ElasticNet) reach dev RMSE between 0.891 and 0.973. The dev-set advantage of the learned XGBoost blend *does not transfer to test*: Blend XGB underperforms the simpler Nelder-Mead and ElasticNet strategies on the hidden test set, and was therefore excluded from our two submissions. We read this as meta-model overfitting to the 677 dev items used to train the

blend — a learned non-linear stacker has enough capacity to absorb dev-set idiosyncrasies that do not generalise, whereas the constrained weighting strategies cannot. Accordingly, the robust portion of the ensemble gain is the ≈ 0.08 RMSE drop shared by every no-learning and linear-meta strategy, which we attribute to *model diversity* rather than to any particular weighting scheme; the two submitted ensembles exceed the closed-track baseline by 0.28 (Es), 0.36 (De) and 0.32 (Cn) RMSE points on test (Table 4).

Variance of Encoder Results. Across the 11 encoders of Table 3 the cross-architecture spread in RMSE is large – from 0.980 (RemBERT) to 1.364 (CANINE-S), a range of 0.38 points – and similar in shape to the cross-architecture Pearson spread. Per-L1 Pearson columns (where the table reports them) further show that base encoders are remarkably stable across L1s (0.005–0.013 Pearson spread, e.g. XLM-R Large .840/.845/.845), so cross-architecture variance dominates cross-L1 variance by roughly an order of magnitude. NMT augmentation roughly doubles cross-L1 Pearson variance (up to 0.020 for XLM-R Large +NMT: .855/.854/.835) and consistently pulls Chinese down relative to Spanish and German – the English-centric NMT systems contribute complementary signal for the two European L1s but interfere with Chinese. This pattern justifies the ensemble design choice of combining multiple *architectures* rather than multiple seeds of a single model.

Language-Specific Weight Preferences. The Nelder-Mead per-language weights reveal interesting patterns. For Spanish, the NMT-augmented XLM-R Large and RemBERT receive the highest weights; for German, NMT-augmented E5-Large and InfoXLM-Large dominate; for Chinese, the base E5-Large and XLM-R Large are preferred. This suggests that different encoder-L1 combinations capture complementary aspects of difficulty, motivating per-language weight fitting over a global weighting scheme.

Encoder vs. Decoder for Difficulty Prediction. The best encoder reaches RMSE 0.980 (RemBERT) while the best decoder-based ALS checkpoint reaches 1.030 (Pythia 1.4B), a gap of 0.05 RMSE points in favour of encoders. While the ALS models show clean monotonic RMSE scaling – 1.512 $\rightarrow$ 1.362 $\rightarrow$ 1.068 $\rightarrow$ 1.030 from 70M to

1.4B parameters – they do not close the gap, and the encoder approach is both more effective and more computationally efficient for this regression task.

NMT Augmentation Helps Both Individually and in Ensembles. NMT augmentation improves RMSE for three of the four top architectures (Table 3): XLM-R Large (1.085 $\rightarrow$ 0.994), RemBERT (0.980 $\rightarrow$ 0.963) and InfoXLM-Large (1.166 $\rightarrow$ 1.043), with E5-Large the lone exception (1.025 $\rightarrow$ 1.046); Pearson rises monotonically across all four. The 8-encoder ensemble (4 base + 4 NMT) reaches dev RMSE 0.891 (ElasticNet) and 0.902 (Nelder-Mead), improving over the best individual NMT model by 0.07–0.08 RMSE points. Exposing the encoder to multiple English renderings of the same L1 input, produced by NMT systems of varying capacity, likely surfaces complementary aspects of crosslinguistic transfer. We did not separately evaluate a base-only (4 encoder) ensemble, so we cannot directly attribute the ensemble gain to the NMT variants alone vs. to model diversity in general.

6 Conclusion

We presented the *uogal* system, which placed 2nd in the closed track of the BEA 2026 shared task on L1-aware vocabulary difficulty prediction. Our approach combines eight fine-tuned multilingual encoders—four base architectures and four NMT-augmented variants—in an ensemble with two stacking strategies, achieving test set Pearson correlations of $r = 0.859$ (Es), $r = 0.869$ (De), and $r = 0.880$ (Cn). We additionally demonstrated monotonic scaling behavior in decoder-based artificial learner simulation, though this approach did not match encoder ensemble performance. Our results indicate that large multilingual encoders can effectively capture L1-specific vocabulary difficulty patterns without explicit L1 conditioning, and that NMT augmentation provides valuable ensemble diversity.

7 Future Work

There are numerous avenues to follow from this work. One direction concerns metadata-aware models, where the L1 identifier is provided explicitly rather than left implicit in the input: our early L1-concat experiments suggest that naive injection interferes with the encoder’s multilingual represen-

tations, and open questions remain about how and where to introduce the L1 signal, whether through token placement, embedding initialization, prefix tuning, or adapter-style conditioning. A second direction is scaling, both in encoder backbones and in the artificial learner simulation side, where the observed trends have not yet plateaued and larger checkpoints or alternative decoder families may narrow the gap to encoder-based difficulty prediction. A third direction is improving the NMT augmentation pipeline by asking a more targeted question: given an L1 context in which a learner encountered a word, what is the best equivalent L2 context to generate? Better answers to this question, whether through context-aware translation, retrieval of comparable L2 passages, or learner-conditioned generation, could make augmentation a stronger signal for L1-specific vocabulary difficulty.

Limitations

Our system has several limitations. First, it was evaluated on only three L1 groups (Spanish, German, Mandarin Chinese); performance on typologically more distant or lower-resource L1 backgrounds remains unknown. Second, the ensemble of eight large models (~ 560 M parameters each) has substantial computational requirements for both training and inference, limiting practical deployment. Third, our ensemble weights were optimized on the development set (677 items per language), which, while a favorable parameter-to-sample ratio ($\sim 85:1$), carries some risk of overfitting. Finally, we report development-set performance for ablation studies (e.g., individual encoder comparison, ALS scaling); only the final ensemble submissions were evaluated on the official test set.

Acknowledgments

This work has been supported by Research Ireland under Grant Number SFI12RC2289_P2 Insight_2, Insight Research Ireland Centre for Data Analytics.

References

Tatsuya Aoyama and Nathan Schneider. 2024. [Modeling nonnative sentence processing with L2 language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4927–4940, Miami, Florida, USA. Association for Computational Linguistics.

- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usven Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
- British Council. 2026. BEA 2026 shared task: Vocabulary difficulty prediction for English learners. <https://www.britishcouncil.org/data-science-and-insights/beat2026st>. Data Science and Insights, British Council. Accessed 2026-04-22.
- 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.
- 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.
- Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. [InfoXLM: An information-theoretic framework for cross-lingual language model pre-training](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.
- Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. [Improving pretrained cross-lingual language models via self-labeled word alignment](#). 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 3418–3430.
- Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.
- Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. [Canine: Pre-training an efficient tokenization-free encoder for language representation](#). *Transactions of the Association for Computational Linguistics*, 10:73–91.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 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.
- Matthew S. Dryer and Martin Haspelmath, editors. 2013. [WALS Online \(v2020.4\)](#). Zenodo.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- 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 (SLRF)*.
- 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*.
- Jingying Hu and Yan Cong. 2025. [Modeling Chinese L2 writing development: The LLM-surprisal perspective](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 172–183, Albuquerque, New Mexico, USA. Association for Computational Linguistics.
- Scott Jarvis and Aneta Pavlenko. 2008. [Crosslinguistic influence in language and cognition](#), 1 edition. Routledge, New York.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30.
- Batia Laufer. 1997. The lexical plight in second language reading: Words you don’t know, words you think you know, and words you can’t guess. In James Coady and Thomas Huckin, editors, *Second Language Vocabulary Acquisition*, pages 20–34. Cambridge University Press.
- 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*.

- David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. [Pan-Phon: A resource for mapping IPA segments to articulatory feature vectors](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484.
- John A. Nelder and Roger Mead. 1965. [A simplex method for function minimization](#). *The Computer Journal*, 7(4):308–313.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Celebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Baez, Gabriel Birber, Johan Boscolo, and 51 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- 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.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. [Scikit-learn: Machine learning in python](#). *Journal of Machine Learning Research*, 12(85):2825–2830.
- 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*, pages 3982–3992.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. [Introducing knowledge-based vocabulary lists \(KVL\)](#). *TESOL Journal*, 12(4):e622.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Robyn Speer. 2022. [wordfreq: v3.0](#).
- 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.
- Bernardo Stearns, Nicolas Ballier, Thomas Gaillat, Andrew Simpkin, and John P. McCrae. 2024. [Evaluating the generalisation of an artificial learner](#). In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 199–208, Rennes, France. LiU Electronic Press.
- Jörg Tiedemann and Santhosh Thottingal. 2020. [OPUS-MT – building open translation services for the world](#). In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. [SciPy 1.0: Fundamental algorithms for scientific computing in Python](#). *Nature Methods*, 17:261–272.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- 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.
- Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Configuration	r	RMSE
all_concat_als_xgb	.797	1.099
all_features_lgbm	.791	1.110
all_features_xgb	.790	1.112
all_concat_ipa_xgb	.782	1.133
all_features_full_xgb	.778	1.140
all_ipa_phonetic_xgb	.777	1.141
cvo_allconcat_xgb	.756	1.188
all_features_lr	.750	1.201
cvo_allconcat_ridge	.738	1.224
cvo_embedding_ridge	.689	1.316
als_nonmt_pythia410m_xgb	.618	1.429
cefr_wordnet_only_xgb	.508	1.563
ipa_phonetic_only_xgb	.507	1.566
cognate_xgb	.404	1.660
nmt_xgb	.385	1.673

Table 5: Detailed dev-set Pearson r and RMSE for the top 15 tabular / scikit-learn configurations over hand-crafted feature modules. Naming convention: {feature_subset}_{regressor}. The best configuration combines all feature modules with ALS-derived embeddings and an XGBoost regressor.

A Detailed Tabular Feature Results

Table 5 reports the dev-set performance of the strongest 15 hand-crafted-feature scikit-learn configurations. All runs use the feature modules described in Section 3.5 (Embeddings, Language-pair typology, IPA phonetic, CEFR + WordNet, Other) in various subsets, paired with five regressor families (linear regression, ridge, decision tree, XGBoost, LightGBM). Approximately 25 additional configurations with $r < 0.40$ (e.g., word-count-only, language-pair-only, single-feature ablations) are omitted for brevity; complete metrics are available in the project repository.

B GPT-2 Decoder Baselines

As a pre-ALS sanity check we fine-tuned native-English GPT-2 decoders (without the EFCAMDAT artificial-learner adaptation) on the same regression head and serialisation described in Section 3.4. These numbers serve purely as a reference point: a generic next-token model with no learner adaptation. They fall well below every multilingual encoder in the main table and below all Pythia-based ALS checkpoints from 410M upwards (Table 3), supporting the design choice to adapt decoders on learner data rather than use them off-the-shelf.

C Full Ensemble-Strategy Ablation

Table 7 reports the full set of ensemble weighting strategies we evaluated on the development

System	Es	De	Cn	Avg r	RMSE
GPT-2	.578	.609	.652	.610	1.476
GPT-2 Medium	.577	.601	.646	.606	1.483

Table 6: GPT-2 decoder baselines on the 677-item dev set. Neither model was adapted on learner data; they serve as a reference for the gain provided by EFCAMDAT-based artificial learner simulation (Table 3).

set. In the main paper (Table 3) we show only the two submitted configurations (Nelder-Mead and ElasticNet meta-learner over the 8 encoders). The remaining rows below include alternative no-learning schemes (simple average, weighted-RMSE, weighted-Pearson, hill climbing, rank average), the learned non-linear Blend XGB meta-model, and a 10-model variant that adds the two strongest tabular configurations (all_features_xgb, all_features_lgbm) to the 8 encoders.

Strategy	Es	De	Cn	Avg r	RMSE
<i>8-encoder ensembles (dev)</i>					
Simple avg	.872	.870	.867	.870	0.973
Weighted RMSE	.872	.871	.867	.870	0.969
Weighted Pearson	.872	.871	.867	.870	0.970
Hill climb	.874	.869	.867	.870	0.910
Rank avg	.868	.871	.860	.866	1.520
Nelder-Mead (submitted)	.875	.870	.868	.871	0.902
ElasticNet meta (submitted)	.875	.870	.867	.871	0.891
Blend XGB	.885	.887	.875	.882	0.851
<i>10-model ensembles — 8 encoders + 2 tabular (dev)</i>					
Nelder-Mead	.878	.870	.868	.872	0.899
ElasticNet meta (dmeta)	.874	.872	.867	.871	0.906
Blend ElasticNet	.873	.873	.868	.871	0.891

Table 7: Full ensemble-strategy ablation on the 677-item dev set. The two submitted configurations (Nelder-Mead and ElasticNet meta-learner over 8 encoders) also appear in Table 3 of the main text. No-learning strategies cluster tightly at $r \in [.866, .871]$; the learned non-linear Blend XGB meta-model reaches the highest average Pearson ($r = .882$) on dev but was not selected for submission.

Jinnie’s Lab at BEA 2026 Shared Task 1: Precalibration of Vocabulary Item Difficulty with Multilingual Transformers and Multi-Task Learning

Zhe Li

University of Florida
Gainesville, FL
zheli@ufl.edu

Pauline Aguinalde

University of Florida
Gainesville, FL
aaguinalde@ufl.edu

Jinnie Shin

University of Florida
Gainesville, FL
jinnie.shin@coe.ufl.edu

Abstract

This paper describes our submission to the BEA 2026 shared task 1 on vocabulary item difficulty prediction in multilingual settings. We investigated whether transformer-based representations learned directly from item content can support the prediction of vocabulary item difficulty across different L1 groups. Our approach adopted a multilingual BERT-based architecture, specifically the mmBERT, with representation augmentation at both the layer and token levels, followed by a multi-task cascade learning that incorporates part-of-speech information as an auxiliary structural signal. Results showed that multi-task mmBERT consistently outperforms the shared-task XLM-RoBERTa baseline across languages, while gains from more complex aggregation are not uniform. The findings showed that strong multilingual representations provide a competitive foundation for vocabulary item difficulty prediction, while the benefits of additional architectural complexity depend on the language and training setting.

1 Introduction

Item difficulty prediction aims to estimate item parameters from item features, with the long-term goal of reducing reliance on resource-intensive calibration procedures while preserving the quality of measurement (Wauters et al., 2012). Within the educational measurement community, this task is not only a question of predictive accuracy, but also of whether model-based estimates can serve as valid and interpretable proxies for empirically calibrated item parameters (e.g., Ulitzsch et al., 2026).

Recent advances in machine learning and natural language processing (NLP), particularly transformer-based models and large language models (LLMs), have substantially expanded the methodological toolkit available for this task, most often through supervised deep learning approaches.

Broadly, existing approaches to item difficulty prediction can be organized along three dimensions: (1) the use of representations learned directly from item text, (2) the incorporation of features generated by LLM-based or other generative frameworks, and (3) the inclusion of response-based information (e.g., student correctness) as part of the prediction process. In this study, we focus specifically on model classes that leverage representations learned directly from item content, treating these representations as the primary source of information for predicting item difficulty.

Transformer-based models are particularly well-suited for this setting due to their ability to generate contextualized, high-dimensional representations of text that capture semantic and syntactic relationships. Their use in item difficulty prediction implicitly assumes that the linguistic signal contained in the item, such as the target word, its context, and surrounding cues, encodes sufficient information to infer the cognitive demand required for a correct response (e.g., Peters et al., 2025). While this assumption has shown promise in domains where item stimuli are relatively self-contained (e.g., reading comprehension), its applicability to more narrowly defined constructs, such as vocabulary knowledge, raises distinct methodological and measurement considerations.

In vocabulary assessment, item difficulty is closely tied to lexical complexity, which reflects not only properties of the target word but also the interaction between the word, its context, and the examinee’s linguistic background. Earlier approaches to vocabulary item difficulty prediction relied heavily on hand-crafted features, including word length, frequency, part-of-speech (POS) tags, concreteness, and age-of-acquisition (Yang and Suyong, 2018; Settles et al., 2020). While these features offer interpretability, they are limited in their ability to capture contextual meaning and semantic variability across usages.

More recent work has shifted toward leveraging contextual embeddings from pre-trained language models, particularly encoder-only architectures such as BERT (Devlin et al., 2019), to model lexical complexity in context (e.g., Zaharia et al., 2021; Kelious et al., 2024). These approaches often combined contextual representations with psycholinguistic features to improve performance. However, from a measurement perspective, it remains unclear to what extent these representations capture construct-relevant variances, as opposed to incidental properties of language use. This challenge becomes more pronounced in cross-lingual settings, where item difficulty may vary as a function of examinees’ first-language (L1) backgrounds and the linguistic relationship between source and target languages. For example, Skidmore et al. (2025) demonstrated that multilingual transformer models (e.g., XLM-RoBERTa Conneau et al., 2020) can be fine-tuned to predict vocabulary item difficulty across multiple languages, suggesting the potential of cross-lingual transfer for item calibration. At the same time, these approaches face challenges related to polysemy, low-frequency cognates, and subword tokenization, all of which may introduce construct-irrelevant variance into predictions.

In this paper, we participate in the BEA 2026 shared task on vocabulary item difficulty prediction and investigate how representation learning and multi-task modeling choices affect predictive performance across languages. Specifically, we explore a multilingual BERT-based architecture (mmBERT; Marone et al., 2025) with layer-wise and token-wise representation aggregation, alongside a multi-task cascade that incorporates part-of-speech (POS) information as an auxiliary structural signal.

Our goal is twofold: (1) to evaluate the effectiveness of these design choices within the shared-task setting, and (2) to examine, at a high level, how different representation strategies relate to the modeling of lexical item difficulty. Through this analysis, we aim to provide both a competitive system and a set of practical insights into what types of representations are most effective for vocabulary item difficulty prediction in multilingual contexts.

2 Related Work

2.1 Transformer-based Item Difficulty Prediction

Within educational measurement, item difficulty is a fundamental parameter that underpins score

interpretation, test assembly, and the evaluation of measurement quality. In both Classical Test Theory (CTT) and Item Response Theory (IRT), difficulty is not directly observed but inferred from examinee response data through calibration procedures. Although these approaches provide a rigorous and defensible basis for parameter estimation and validity arguments, they rely on sufficiently large and representative samples, which can limit the efficiency and scalability of item development. Consequently, a growing body of research has examined the prediction of item difficulty prior to operational calibration using features derived from item content (e.g., Peters et al., 2025). These efforts are best understood not as substitutes for psychometric calibration, but as methods for producing provisional estimates that can inform item screening, test construction, and adaptive content generation in the absence of response data.

Recent advances in transformer-based models and large language models (LLMs) have expanded the methodological approaches available for this purpose. One class of methods uses pretrained language models to derive semantic representations of item text, which are then used to predict relative or absolute difficulty. For example, Loginova et al. (2021) demonstrated that contextual embeddings can support pairwise difficulty comparisons without relying on calibrated parameters or observed response vectors, although such predictions remain model-based approximations rather than psychometric estimates grounded in examinee behavior. Related work (e.g., AlKhuyaey et al., 2024; McCarthy et al., 2021) similarly treats text as the primary source of information for difficulty prediction.

A second line of work attempts to approximate difficulty through simulated response processes, often by treating LLMs as proxy examinees. Maeda (2025) generate synthetic response data from trained models to estimate item difficulty in settings where field testing is limited. While promising, these approaches introduce additional assumptions regarding the correspondence between model-generated responses and human performance, raising important questions about construct representation, population invariance, and the interpretability of resulting parameters.

2.2 Vocabulary Item Difficulty Prediction

A specific area within item difficulty prediction concerns vocabulary assessment, where the item difficulty is closely tied to the linguistic properties

of target words. In these contexts, predicting difficulty prior to calibration often relies on features that capture how learners process lexical content.

A closely related line of work is lexical complexity prediction, which reflects the perceived difficulty of a word for a given population (North et al., 2023). This construct has been widely studied in NLP and applied linguistics, particularly in tasks such as lexical simplification (Shardlow et al., 2022). In vocabulary learning contexts, complexity is typically modeled using morphological, semantic, and contextual features, sometimes incorporating learner characteristics such as L1. For example, Alfter and Volodina (2018) predicted comprehension levels for Swedish learners using linguistically informed features and machine learning methods. More recent work has leveraged contextualized representations from transformer-based models (e.g., BERT, RoBERTa), showing improved performance over traditional approaches such as SVM and Random Forest (Zaharia et al., 2021; Yaseen et al., 2021).

From a measurement perspective, lexical complexity and item difficulty are conceptually distinct. Lexical complexity operates at the word level and reflects processing demands, whereas item difficulty is a latent parameter defined through examinee responses. Nonetheless, lexical complexity provides a construct-relevant source of information for vocabulary item difficulty prediction. Word-level features may explain meaningful variation in item difficulty, particularly when aligned with population characteristics such as learners’ L1, which can systematically influence lexical processing (Graham et al., 2010; Derakhshan and Karimi, 2015; Schneider et al., 2026).

2.3 Multilingual BERT regression

Predicting item difficulty that accounts for lexical complexity can be approached as a regression problem. As such, transformer-based regression approaches have been adopted to predict lexical complexity. Ortiz-Zambrano et al. (2025)’s study highlighted the utility of incorporating linguistic features with BERT, XLMR, and RoBERTa towards improving lexical complexity prediction across English and Spanish contexts. Variations of transformer-based models employing regression models have been introduced in a lexical complexity prediction task in 2021 (North et al., 2023). DeepBlueAI, one of the top performing models, incorporated a final estimator with a simple linear

regression model (Pan et al., 2021).

Similarly, BERT-based regression models have been employed to predict item difficulty (Yaneva et al., 2024). Benedetto et al. (2023) analyzed quantitative approaches to item difficulty and found that BERT consistently outperforms baseline machine learning models, while hybrid models that incorporate linguistic features further improve prediction performance. Further studies have adopted regression-based approaches for difficulty prediction within various domains, including medicine (Ram and Kesanam, 2024), algebra (Aradelli, 2019), and reading (Kapounová, 2025). Foundational to this year’s BEA shared task, Skidmore et al. (2025) examined item difficulty in a vocabulary learning setting. Thus, integrating insights from research on lexical complexity and item difficulty prediction are promising towards modeling difficulty in multilingual learning contexts.

2.4 Research Gap

Despite progress in lexical item complexity prediction, estimating the continuous difficulty of vocabulary test items in cross-lingual contexts remains difficult. Traditional methods emphasize hand-crafted linguistic features but fail to capture deep semantic dependencies (Yang and Suyong, 2018; Settles et al., 2020), while standard multilingual transformers offer cross-lingual transfer yet often miss nuanced difficulty variations confounded by specific target word senses (Skidmore et al., 2025). Although some work proposes integrating psycholinguistic features with deep embeddings, most automated approaches merely concatenate deep semantic representations with hand-crafted features prior to final prediction (Bulut et al., 2024; Benedetto et al., 2023; Ortiz-Zambrano et al., 2025). Furthermore, advanced frameworks such as multi-task learning, which leverage auxiliary linguistic tasks to improve primary predictions (Sanh et al., 2018; Liu et al., 2019; Galal et al., 2024), remain underexplored. To address this gap, we introduce an mmBERT-based multi-task architecture with layer-wise and token-wise aggregation that explicitly models cross-lingual semantic relationships and syntactic biases through auxiliary POS tagging, enabling more accurate and context-aware difficulty prediction.

3 Methods

In this section, we present the system architecture, including the data preprocessing, BERT-based rep-

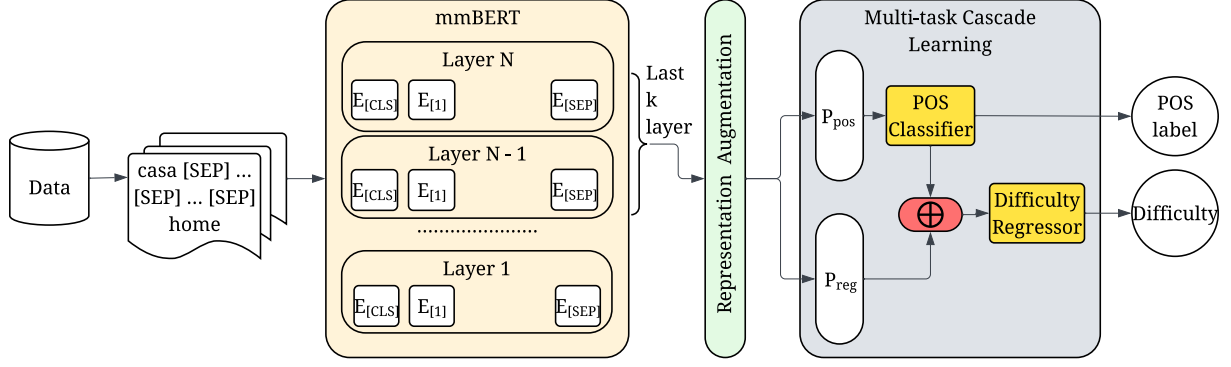


Figure 1: Overview of the proposed system architecture.

representation generation, the layer-wise and token-wise representation augmentation, and multi-task downstream prediction layer. An overview of the system is presented in Figure 1. First, the item text is tokenized by concatenating distinct linguistic components into a single multilingual sequence. Second, this sequence is passed through a BERT family model, specifically the mmBERT architecture, to extract comprehensive cross-lingual semantic representations. To maximize the utility of the deep semantic information, the system applies both layer-wise fusion, such as scalar mixing or mean/max token pooling, and token-wise aggregation via self-attention to output an augmented item representation. Third, the augmented representation is utilized in a hierarchical multi-task architecture. It initially computes auxiliary part-of-speech tagging logits for the English target word. Then, these auxiliary signals are detached and injected into the primary task by concatenating directly with the primary semantic representation to predict the final continuous lexical difficulty score.

To encode the test item, we followed the approach by Skidmore et al. (2025) to organize the input text into a single multilingual sequence. The sequence follows the presentation order of the vocabulary test items, concatenating the source language word, the source language context, the English target clue, and the target English word using the [SEP] tokens. This structure allows the encoder to capture cross-lingual relationships and contextual constraints.

casa [SEP] Vivo en una casa grande
que tiene tres dormitorios. [SEP]
h_____ [SEP] home

Target labels In this supervised framework, each task utilizes a label to guide learning. The primary label is a continuous difficulty score derived from generalized linear mixed model (GLMM) estimates to quantify the lexical challenge of the English target word. Concurrently, the system predicts an auxiliary label representing the POS tag of the target word. Because English target words are lemmas agnostic to specific word senses, and certain senses are more difficult for learners, the POS tag provides a structural inductive bias. This is directly injected into the final difficulty predictor through a multi-task cascading architecture (Goldberg, 2022).

Representation Generation Given BERT-based regression model’s strong performance and widespread success on both lexical complexity prediction and item difficulty prediction tasks (Ortiz-Zambrano et al., 2025; Ram and Kesanam, 2024; Kelious et al., 2024; Skidmore et al., 2025; Kapounová, 2025), our system uses the encoder-only model to serve as the representation generator. Specifically, the system utilizes mmBERT (Marone et al., 2025), the latest multilingual encoder that excels across diverse multilingual NLP tasks compared to its predecessors (e.g. XLM-RoBERTa).

3.1 Representation augmentation

After obtaining the token-level representation of the item sequence using the mmBERT encoder, we aggregate hidden states across both intermediate layers and sequence tokens, inspired by the finding that task-relevant information in transformers is distributed within its deep and wide structure (Rogers et al., 2021; Galal et al., 2024; Behrendt et al., 2025; Ciernik et al., 2026). For layer-wise aggregation, we experimented with three approaches separately: scalar mixing, mean pooling, and max pooling. The last several layers are selected to prevent contam-

ination of irrelevant information encoded in the lower layer. Once we get an aggregated layer representation, we utilize either the self-attention mechanism or token-wise mean pooling to expand the sentence head into a sequence-wide representation.

Given a BERT-based encoder with T number of intermediate layers, we write $\mathbf{H} \in \mathbb{R}^{k \times l \times d}$ for the tensor of item sequence’s hidden states of multiple layers, where k is the number of chosen layers, l is the sequence length, and d is the hidden size. Let $h_i \in \mathbb{R}^{l \times d}$ denote the sequence representation at the i -th layer.

Scalar Mixing Learn softmax weights s_i to dynamically compute a weighted sum of hidden states h_i across k layers of the network, where γ is a trainable scaling factor.

$$\mathbf{H}_{mix} = \gamma \sum_{i=T-k+1}^T \text{softmax}(s_i) \cdot h_i$$

For each layer, a layer normalization (Ba et al., 2016) is used to respect the fact that their activations might distribute differently to address certain aspects of the task. Specifically, we adopted PyTorch’s `layer_norm()` on each layer $h_i = \text{LN}(h_i)$. After the contextually mixed representation $\mathbf{H}_{mix}$ is generated, we apply the attention mask to zero-out padding tokens in each sequence that would corrupt the loss signals.

Mean/Max layer pooling Apply a deterministic pooling operation across the depth dimension to extract the element-wise average or maximum values. Specifically, we aggregate the hidden states from the k chosen layers of the network:

$$\mathbf{H}_{mean} = \frac{1}{k} \sum_{i=T-k+1}^T h_i, \quad \mathbf{H}_{max} = \max_{T-k+1 \leq i \leq T} h_i$$

We set $k = 4$ to aggregate the deep semantic information captured in the final four encoder layers.

Self-attention To integrate sequence-wide contextual information into the global representation, we extract the [CLS] token $q_{cls} \in \mathbb{R}^{1 \times d}$ from the layer-aggregated sequence $\mathbf{H}^{(1)} \in \mathbb{R}^{l \times d}$ to serve as the single query, while the entire sequence $\mathbf{H}^{(1)}$ serves as the keys and values. We write self-attention aggregated representation $\mathbf{h}_{attn} \in \mathbb{R}^d$ as:

$$\begin{aligned} \mathbf{h}_{attn} &= \text{Attention}(q_{cls}, \mathbf{H}^{(1)}, \mathbf{H}^{(1)}) \\ &= \text{Weight} \cdot \mathbf{H}^{(1)} \cdot W_V \end{aligned}$$

where the attention weights are defined as

$$\text{Weight} = \text{softmax} \left(\frac{(q_{cls} \cdot W_Q)(\mathbf{H}^{(1)} \cdot W_K)^T}{\sqrt{d}} \right)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are trainable projection matrices. Here, $\text{Weight} \in \mathbb{R}^{1 \times l}$ represents the attention weights between the [CLS] token and each token in the item sequence. This mechanism enables the global representation to selectively attend to relevant segments of the source, context, and clue components, producing a contextualized representation that informs the final prediction.

Mean token pooling Same as the depth-wise mean pooling, it applies a deterministic token-wise mean pooling operation that aggregates the sequence into a unified representation. It computes the element-wise average of the hidden states across all existing tokens in the sequence. A binary attention mask $m \in \{0, 1\}^l$ is applied to filter out padding tokens, where $m_j = 1$ for valid tokens and 0 for padded tokens. The mean-pooled sequence representation $\mathbf{h}_{mean} \in \mathbb{R}^d$ is given as:

$$\mathbf{h}_{mean} = \frac{1}{\sum_{j=1}^l m_j} \sum_{j=1}^l m_j \cdot h_j$$

3.2 Multi-task Cascade Learning

To capture the interdependency between syntax and lexical complexity, the system employs a multi-task learning (MTL) strategy utilizing hard parameter sharing. Instead of concatenating the one-hot POS labels with continuous deep representation, which introduces numerical incompatibility, our model is jointly supervised by the primary difficulty prediction task and the auxiliary POS tagging task. In this way, the auxiliary task training loss introduces a strong inductive bias for regularizing the training of shared hidden layers (Liu et al., 2019).

Furthermore, rather than deploying two isolated task heads on top of the shared hidden layers, our system utilizes a cascade architecture for hierarchical feature injection (Sanh et al., 2018; Aziz et al.). The model first computes a C -dimensional real-valued POS logit vector $\hat{y}_{pos}$ from the auxiliary classification head, where C represents the number of POS classes. To prevent the primary continuous regression loss from introducing noisy interference into the auxiliary syntactic head, these logits are detached from the primary gradient computation. The detached syntactic signals are then concatenated with the augmented deep representation for

the regressor, p_{reg} , to form the combined vector for the final difficulty prediction:

$$X_{combined} = [p_{reg} \oplus \text{detach}(\hat{y}_{pos})]$$

Note that while both tasks share the same backbone hidden layers, their additional augmentation modules are optimized independently. Consequently, the augmented deep representation utilized for POS classification, p_{pos} , remains distinct from the regression representation p_{reg} . This structural separation allows each head to specialize in its respective objective while ensuring the learned syntactic signals act as a fixed structural hint to guide the primary difficulty estimation.

Dynamic loss For the primary difficulty regression task, we calculate the mean squared error loss L_{reg} , while applying cross-entropy loss L_{pos} for the auxiliary POS tagging task. Because these two tasks operate on different scales, a naive summation would cause one objective to dominate the gradient updates. To jointly optimize the network without the need of extensive manual search for task weights, we employ dynamic precision weighting based on homoscedastic task uncertainty (Kendall et al., 2018). The total loss adaptively scales each task’s contribution:

$$\mathcal{L}_{total} = \frac{1}{2\sigma_0^2}\mathcal{L}_{reg} + \log \sigma_0 + \frac{1}{\sigma_1^2}\mathcal{L}_{pos} + \log \sigma_1$$

where σ_0 and σ_1 represent the task-dependent observation noise. In practice, we utilize the log-variance of each task to avoid zero division error.

4 Experiments

4.1 Dataset

The dataset for the shared task is taken from the "Extended KVL Dataset for NLP" released by Skidmore et al. (2025) and is organized into two independent tracks with specific data constraints.

Closed Task Contains 6,091 training, 677 development, and 748 test items for each of three source languages (German, Spanish, and Mandarin). Each item is associated with the POS label of the target English word and its GLMM score. Systems in this track are restricted to using L1-specific training data and publicly available pre-trained transformer models and their embeddings.

Open Task Permits the integration of all available data, allowing the combination of the three L1 subsets into a unified training set of 18,273 items.

Systems may train on this joint training set and evaluate on the L1-specific development set and test set. No algorithmic or architectural restrictions are imposed. Because the proposed system relies exclusively on the publicly available mmBERT encoder for representation generation, it complies with the constraints of both tracks.

4.2 Setup

We fine-tuned all models utilizing the mmBERT-base architecture, which contains roughly 307M parameters. Inputs are processed using the Gemma 2 tokenizer, which is a native implementation employed by mmBERT to support multilingual pre-training (Marone et al., 2025). The augmented classification token from the sequence representation serves as the input for both the classification and regression tasks. Specifically, this 768-dimensional hidden representation is passed into a two-layer MLP prediction head consisting of an initial linear layer, a GELU activation, layer normalization, dropout, and a final linear layer to generate the output logits. Our baseline employs the ordinary single-task MLP regressor which uses only the representation of the first token of the last hidden layer on top of mmBERT and XLM-RoBERTa base.

All models were trained for 10 epochs using the AdamW optimizer with a fixed learning rate of $2e-5$, a weight decay of 0.1, and 100 warmup steps, with a batch size of 64 and a dropout rate of 0.1 throughout training. The best model checkpoint was selected and saved based on the lowest RMSE on the development set, with Pearson’s correlation coefficient (ρ) reported as a supplementary evaluation metric. The system was trained and validated on the provided training and development sets, and the top three performing configurations were subsequently selected for hyperparameter optimization before generating final test set predictions for each L1 as the submission (See Table 6 in the Appendix).

5 Results

This section presents the evaluation results of our mmBERT-based multi-task cascade models (MTL) for the shared task. We present four specific augmentation configurations: self-attention token aggregation (MTL+SelfAttn), mean token pooling (MTL+MeanToken), scalar mixing layer aggregation followed by self-attention token aggregation (MTL+ScalarMix+SelfAttn), and mean layer pooling followed by mean token pooling (MTL+MeanLayer+MeanToken). They were eval-

Table 1: Model performance overview in the open track.

Model	ES		DE		CN		L1 avg	
	RMSE	ρ	RMSE	ρ	RMSE	ρ	RMSE	ρ
XLM-RoBERTa base	1.198	0.783	1.166	0.786	1.034	0.804	1.133	0.791
mmBERT base	1.078	0.820	1.014	0.826	0.910	0.842	1.000	0.829
MTL	1.081	0.819	1.011	0.826	0.910	0.842	1.000	0.829
MTL+SelfAttn	1.068	0.824	1.002	0.830	0.906	0.843	0.992	0.832
MTL+MeanToken	1.066	0.824	1.001	0.830	0.910	0.841	0.993	0.832
MTL+ScalarMix+SelfAttn	1.153	0.792	1.067	0.808	0.968	0.819	1.063	0.806
MTL+MeanLayer+MeanToken	1.087	0.817	1.018	0.824	0.910	0.842	1.005	0.828

Table 2: Model performance overview in the closed track.

Model	ES		DE		CN		L1 avg	
	RMSE	ρ	RMSE	ρ	RMSE	ρ	RMSE	ρ
XLM-RoBERTa base	1.257	0.765	1.258	0.773	1.140	0.753	1.218	0.764
mmBERT base	1.138	0.797	1.069	0.804	0.950	0.825	1.033	0.812
MTL	1.126	0.802	1.058	0.809	0.936	0.832	1.020	0.817
MTL+SelfAttn	1.120	0.805	1.059	0.810	0.939	0.830	1.034	0.816
MTL+MeanToken	1.112	0.801	1.050	0.812	0.942	0.829	1.035	0.814
MTL+ScalarMix+SelfAttn	1.325	0.715	1.213	0.740	0.977	0.815	1.172	0.756
MTL+MeanLayer+MeanToken	1.117	0.805	1.061	0.810	0.943	0.830	1.028	0.816

uated on the German (DE), Spanish (ES), and Mandarin (CN) test sets across both the closed and open tracks, and compared against the reference baselines (Section 4.2). Results are summarized in Tables 1 and 2 for the open and closed tracks, respectively. Detailed comparisons across all configurations are provided in Table 4 and Table 5 in the Appendix for both tracks.

Our top configurations ranked 13th/24 for Spanish and German and 14th/26 for Mandarin in the open track, and 16th/56 for Spanish and 13th/53 for both German and Mandarin in the closed track. As expected, performance in the open track consistently exceeded that of the closed track across all languages, reflecting the advantage of access to larger and combined training data.

Across both tracks, models built on the mmBERT encoder consistently outperformed the XLM-RoBERTa baseline, indicating that improved multilingual representations provide a strong foundation for item difficulty prediction. In particular, the base mmBERT model already yielded substantial gains, with further improvements observed when incorporating selected representation aggregation strategies. Among these aggregations, methods such as self-attention token aggregation and mean layer pooling plus mean token pooling yielded consistent improvements, with the former excelling in the open track and the latter in the

closed track. The mean token pooling produced comparable results. The only exception was the most complex mechanism (self-attention-based aggregation, scalar mixing). It led to performance degradation in almost all languages, particularly in the closed track.

The impact of multi-task learning with auxiliary POS prediction consistently improved the results across languages and tracks, and performance in the closed track was boosted more prominently than the open track. It suggests that syntactic information provided useful signals that contribute to the lexical difficulty prediction, with the degree of improvement depending on the data conditions. Lastly, performance patterns varied across languages. Mandarin generally exhibited larger gains from representation learning and aggregation strategies, whereas results for German and Spanish were more stable but showed smaller improvements.

6 Discussion

6.1 Effect of the mmBERT Encoder

Table 1 and 2 show that fine-tuning mmBERT encoder yielded substantially better performance than the standard XLM-RoBERTa encoder baseline across almost all languages and data conditions. In the open track, replacing the XLM-RoBERTa backbone with mmBERT reduced the average RMSE

from 1.133 to 1.042, and increased the average ρ from 0.791 to 0.815. The performance gain largely persisted in the restricted data condition of the closed track, where mmBERT reduced the overall average RMSE from 1.218 to 1.033 and increased average ρ from 0.764 to 0.812.

As the multilingual adaptation of the ModernBERT (Warner et al., 2025), mmBERT is pretrained on the 3T tokens and 1,833 languages with innovative training recipes. Its superior performance over XLM-RoBERTa may be attributed to a combination of modernized architectural enhancements, a more diverse and larger pretraining corpus, and highly optimized data scaling (Marone et al., 2025). Our results showed that mmBERT provides a better alternative to encoder-only models in cross-lingual vocabulary item difficulty prediction.

6.2 Multitask performance

Table 1 shows that in the open track, which provides an expanded joint training set, the multitask approach produced similar performance regarding the difficulty prediction objective. In the restricted data conditions (Table 2), the MTL showed some marginal improvements, with average RMSE decreasing from 1.033 to 1.020, and average ρ increasing from 0.812 to 0.817.

A further benefit of the MTL extension that we observed involves improved robustness in difficulty prediction. As shown in Figure 2, the MTL model demonstrated more consistent performance, particularly under the restricted data condition in the closed track. The incorporation of POS-tagging signals may help the model disambiguate target English words with variable semantic roles, thereby supporting more stable predictions of the word-level item difficulty.

Table 3: Fixed-effects regression results on Δ RMSE.

Predictor	Coef.	SE	<i>t</i>	<i>p</i>	95% CI
Intercept	0.452	0.382	1.183	0.238	[-0.300, 1.204]
$\mathbb{I}(\text{Closed-DE})$	0.001	0.004	0.204	0.838	[-0.008, 0.009]
$\mathbb{I}(\text{Closed-ES})$	-0.000	0.006	-0.009	0.993	[-0.012, 0.012]
$\mathbb{I}(\text{Open-CN})$	0.014	0.004	3.121	0.002	[0.005, 0.022]
$\mathbb{I}(\text{Open-DE})$	0.012	0.004	2.756	0.006	[0.003, 0.020]
$\mathbb{I}(\text{Open-ES})$	0.017	0.004	4.025	< 0.001	[0.009, 0.026]
POS Accuracy	-0.478	0.392	-1.219	0.223	[-1.248, 0.293]

Despite the robustness against semantic noises, accurate POS tagging did not show statistically significant correlations with difficulty prediction improvement. We operationalized the performance improvement as the RMSE difference on the test set

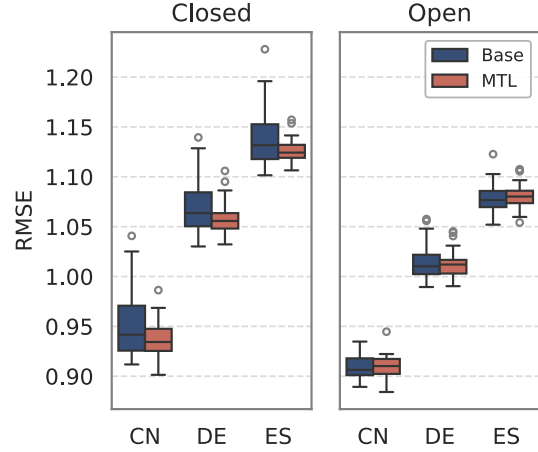


Figure 2: Test RMSE performance between mmBERT base model and its MTL variant across tracks and L1s

between each pair of the MTL model and the base model (i.e. Δ RMSE = $\text{RMSE}_{\text{MTL}} - \text{RMSE}_{\text{Base}}$) across tracks and L1s.

The regression plots in Figure 3 did not reveal a consistent linear relationship between POS prediction accuracy and Δ RMSE across tracks and L1 conditions. Although slight negative trends appeared for some language-track combinations (e.g., CN and DE), the overall patterns were weak and inconsistent, with substantial variability around the fitted regression lines. To further isolate the effect of POS accuracy on RMSE differences, we conducted a fixed-effects linear regression controlling for track and L1 conditions. The results (Table 3) likewise did not support a significant association between POS prediction accuracy and RMSE improvement. One possible explanation is that word-level POS tagging provides only limited linguistic information, making it insufficient to substantially enhance the difficulty prediction objective.

In general, MTL learns a more robust and generalized representation on related tasks by ignoring data-dependent noise and reduces overfitting risks (Ruder, 2017; Goldberg, 2022), which is especially beneficial with restricted data in the closed track. However, the benefit to the primary difficulty prediction objective appears to be limited as the MTL model did not yield significantly better RMSE results than the baseline model. This may be because the auxiliary POS tagging task is relatively simple and may not provide sufficiently rich linguistic information to substantially improve word-level difficulty prediction. It suggested that POS tagging is a related task that can be jointly trained

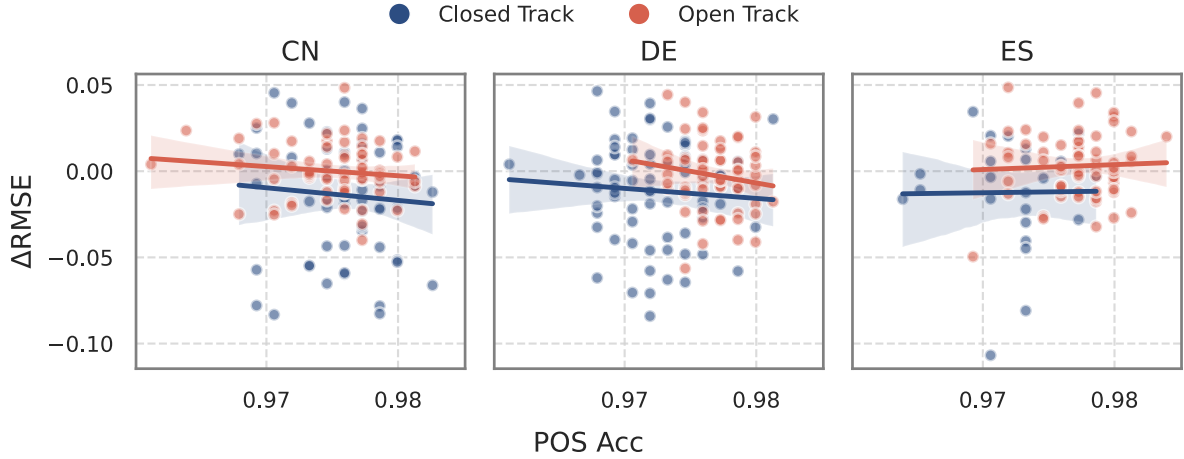


Figure 3: The regression plot between RMSE improvement and POS Acc across tracks and L1s. The x-axis refers to the POS prediction accuracy by MTL model, while the y-axis refers to the difference between RMSEs scored by the MTL model and the base model on the same L1 test set. Color stands for which track has been used as the training set.

with lexical difficulty prediction without introducing conflicting signals. This aligns with the finding that different word senses of a polysemy contribute differently to the difficulty of the vocabulary item (Skidmore et al., 2025).

6.3 Mixed Outcomes by Representation Augmentation

Systematic comparison of configurations revealed mixed outcomes by hidden states aggregation (see Table 4 and Table 5). Prior studies demonstrated that allowing the classification token to attend to the full sequence via an additional multi-head attention module consistently enhances the standard BERT baseline, with strong benefits observed in low-resource scenarios (Behrendt et al., 2025). Consistently, our results indicated that token-wise self-attention aggregation alone provided a slight performance boost, while it severely conflicted with layer-wise aggregation methods. A possible explanation is that naively aggregating intermediate layers through simple deterministic operations, such as mean/max pooling, failed to adapt to task-specific requirements and introduced severe optimization instability during training (Ciernik et al., 2026).

To mitigate this instability, scalar mixing computes a weighted average of hidden layers by dynamically determining the importance of each layer during the training. In theory, it would allow optimal exploitation of deeper information and surpass naive deterministic pooling methods (Gombert et al., 2024). However, our results showed a con-

tradicting image, with the scalar mixing being the worst approach. An investigation into the module showed that all trained scalar weights s collapsed into a delta distribution with a single layer sticking out and all else muted. Also, the scaling factor γ either exploded or vanished. This indicated that the training with scalar mixing virtually failed to learn meaningful weights as expected.

Moreover, the simple mean token pooling approach only trailed the self-attention approach by a small margin. Most importantly, our result indicated that it worked smoothly with other deterministic layer aggregation approaches, especially with limited data. This suggested mean token pooling as a robust and efficient alternative.

7 Conclusions

In this paper, we examined the precalibration of vocabulary item difficulty in multilingual settings using transformer-based representations and multi-task learning. Across both open and closed tracks, our findings showed that mmBERT provided a stronger foundation than the shared-task XLM-RoBERTa baseline. Moreover, the auxiliary POS-based multi-task learning further improved the performance. At the same time, the benefits of additional architectural complexity, including layer-wise and token-wise aggregation, were not uniform across language and data conditions.

8 Limitations

As discussed in section 6.3, while some representation aggregation strategies offered promising results, their improvements over the baseline were context-dependent and not uniform. This indicates that, within the multi-task learning regime for this specific task, there may not be a universally superior representation aggregation strategy. Consequently, any future research seeking to utilize internal encoder states may benefit from extensive and systematic model selection to identify the optimal configuration. Also, the reason behind our failed training with scalar mixing needs to be further researched. Given that the performance gains are marginal and unstable, the standard last-layer [CLS] approach still remained a highly competitive and computationally efficient default.

Second, our study largely relied on a multilingual encoder that provides a language-agnostic foundation for cross-lingual transfer. However, we did not evaluate language-specific monolingual encoders. Previous studies indicated that specialized monolingual models may capture distinct, language-dependent morphological and syntactic patterns more effectively (Ortiz-Zambrano et al., 2025; Skidmore et al., 2025). The absence of experiments using language-specific encoders, such as Chinese ModernBERT (Zhao et al., 2025) or ModernGBERT (Wunderle et al., 2025), limits our understanding of whether specialized pre-training could yield superior lexical difficulty estimations. This limitation is particularly relevant for the closed track, where models were trained and evaluated within each language space.

References

- David Alfter and Elena Volodina. 2018. Towards single word lexical complexity prediction. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 79–88. Association for Computational Linguistics.
- Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2024. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, 34(3):862–914.
- Gokul Aradelli. 2019. Transformers for question difficulty estimation from text.
- Abdul Aziz, Md. Akram Hossain, Abu Nowshad Chy, Md. Zia Ullah, and Masaki Aono. [Leveraging contextual representations with BiLSTM-based regressor for lexical complexity prediction](#). 5:100039.
- Jimmy Ba, J. Kiros, and Geoffrey E. Hinton. 2016. [Layer Normalization](#).
- Maike Behrendt, Stefan Sylvius Wagner, and Stefan Harmeling. 2025. MaxPoolBERT: Enhancing BERT Classification via Layer- and Token-Wise Aggregation. *arXiv preprint*.
- 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.
- 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)*, pages 522–527.
- Laure Ciernik, Marco Morik, Lukas Thede, Luca Eyring, Shinichi Nakajima, Zeynep Akata, and Lukas Muttenthaler. 2026. [Beyond the final layer: Attentive multilayer fusion for vision transformers](#). *arXiv preprint*.
- 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. *arXiv preprint*.
- Ali Derakhshan and Ehsan Karimi. 2015. The interference of first language and second language acquisition. *Theory and Practice in Language Studies*, 5(10):2112–2117.
- 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.
- Omar Galal, Ahmed H. Abdel-Gawad, and Mona Farouk. 2024. Rethinking of BERT sentence embedding for text classification. *Neural Computing and Applications*, 36:20245–20258.
- Yoav Goldberg. 2022. Cascaded, multi-task and semi-supervised learning. In *Neural Network Methods for Natural Language Processing*, pages 235–250. Springer.
- Sebastian Gombert, Lukas Menzel, Daniele Di Mitri, and Hendrik Drachler. 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)*, pages 483–492.

- Charles R. Graham, James McGhee, Brenton Millard, Matthew T. Prior, Yasuko Watanabe, and Sun Lee. 2010. The role of lexical choice in elicited imitation item difficulty. In *Selected Proceedings of the 2008 Second Language Research Forum*, pages 57–72, Somerville, MA. Cascadilla Proceedings Project.
- Ivana Kapounová. 2025. Predicting item difficulty by applying machine learning algorithms using item text features.
- Abdelhak Kelious, Matthieu Constant, and Christophe Coeur. 2024. Investigating strategies for lexical complexity prediction in a multilingual setting using generative language models and supervised approaches. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 96–114.
- Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. [Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics](#). *arXiv preprint*.
- Yifan Liu, Bohan Zhuang, Chunhua Shen, Hao Chen, and Wei Yin. 2019. [Auxiliary Learning for Deep Multi-task Learning](#). *arXiv preprint*.
- Ekaterina Loginova, Luca Benedetto, Damien Benoit, and Paolo Cremonesi. 2021. Towards the application of calibrated transformers to the unsupervised estimation of question difficulty from text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 846–855. INCOMA Ltd.
- Hirofumi Maeda. 2025. Field-testing multiple-choice questions with ai examinees: English grammar items. *Educational and Psychological Measurement*, 85(2):221–244.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.
- Arianna D. McCarthy, Kevin P. Yancey, Gregory T. LaFlair, Jesse Egbert, Mengxuan 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. Association for Computational Linguistics.
- Kathryn North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.
- Juan A. Ortiz-Zambrano, Carlos H. Espín-Riofrío, and Arturo Montejo-Ráez. 2025. Deep encodings vs. linguistic features in lexical complexity prediction. *Neural Computing and Applications*, 37(3):1171–1187.
- Cheng Pan, Bing Song, Shuo Wang, and Zhen 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. Association for Computational Linguistics.
- Sydney Peters, Nan Zhang, Hong Jiao, Ming Li, Tianyi Zhou, and Robert Lissitz. 2025. Text-based approaches to item difficulty modeling in large-scale assessments: A systematic review. *arXiv preprint arXiv:2509.23486*.
- G. V. R. Ram and A. Kesanam. 2024. Leveraging physical and semantic features of text item for difficulty and response time prediction of usmlt questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 534–541. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2018. [A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks](#). *arXiv preprint*.
- Christoph Schneider, Jing Chen, and James Heneger. 2026. Text complexity versus task complexity: Item difficulty modeling for reading items. *Practical Assessment, Research, and Evaluation*, 31(1).
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for computational Linguistics*, 8:247–263.
- 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.
- Lucy Skidmore, Mariano Felice, and Karen J Dunn. 2025. Transformer architectures for vocabulary test item difficulty prediction.
- Esther Ulitzsch, Dmitry Belov, Oliver Luedtke, and Alexander Robitzsch. 2026. Using item parameter predictions for reducing calibration sample requirements—a case study based on a high-stakes admission test. *Journal of Educational Measurement*, 63(1):e12426.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast,

- memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.
- 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.
- Julia Wunderle, Anton Ehrmanntraut, Jan Pfister, Fotis Jannidis, and Andreas Hotho. 2025. New Encoders for German Trained from Scratch: Comparing ModernGBERT with Converted LLM2Vec Models. *arXiv preprint*.
- Victoria Yaneva, Kathryn North, Peter Baldwin, Le An Ha, Sara Rezayi, Yufan Zhou, 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)*, pages 470–482. Association for Computational Linguistics.
- Hua Yang and EUM Suyong. 2018. Feature analysis on english word difficulty by gaussian mixture model. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 191–194. IEEE.
- Talal B. Yaseen, Qusai Ismail, Sara Al-Omari, Eyad Al-Sobh, and Mohammad 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. Association for Computational Linguistics.
- George-Eduard Zaharia, Dumitru-Clementin Cercel, and Mihai Dascalu. 2021. Upb at semeval-2021 task 1: Combining deep learning and hand-crafted features for lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 609–616.
- Zeyu Zhao, Ningtao Wang, Xing Fu, and Yu Cheng. 2025. Chinese ModernBERT with Whole-Word Masking. *arXiv preprint*.

A Appendix

Table 4: Comparison of multi-task mmBERT models with variant augmentations in the open track.

Model	RMSE	ρ
baseline	0.910	0.842
SelfAttn	0.906	0.843
ScalarMix+SelfAttn	0.968	0.819
MeanLayer+SelfAttn	0.938	0.833
MaxLayer+SelfAttn	0.918	0.840
MeanToken	0.910	0.841
ScalarMix+MeanToken	0.980	0.813
MeanLayer+MeanToken	0.910	0.842
MaxLayer+MeanToken	0.918	0.839

(a) Open Track: Mandarin

Model	RMSE	ρ
baseline	1.011	0.827
SelfAttn	1.002	0.830
ScalarMix+SelfAttn	1.067	0.808
MeanLayer+SelfAttn	1.054	0.813
MaxLayer+SelfAttn	1.057	0.814
MeanToken	1.001	0.830
ScalarMix+MeanToken	1.053	0.811
MeanLayer+MeanToken	1.018	0.824
MaxLayer+MeanToken	1.013	0.826

(b) Open track: German

Model	RMSE	ρ
baseline	1.081	0.819
SelfAttn	1.068	0.824
ScalarMix+SelfAttn	1.153	0.792
MeanLayer+SelfAttn	1.141	0.802
MaxLayer+SelfAttn	1.090	0.820
MeanToken	1.066	0.824
ScalarMix+MeanToken	1.134	0.798
MeanLayer+MeanToken	1.087	0.817
MaxLayer+MeanToken	1.078	0.821

(c) Open track: Spanish

Table 5: Comparison of multi-task mmBERT models with variant augmentations in the closed track.

Model	RMSE	ρ
baseline	0.936	0.832
SelfAttn	0.939	0.830
ScalarMix+SelfAttn	0.977	0.815
MeanLayer+SelfAttn	0.963	0.824
MaxLayer+SelfAttn	0.999	0.808
MeanToken	0.942	0.829
ScalarMix+MeanToken	1.088	0.764
MeanLayer+MeanToken	0.943	0.830
MaxLayer+MeanToken	0.952	0.825

(a) Closed Track: Mandarin

Model	RMSE	ρ
baseline	1.058	0.804
SelfAttn	1.059	0.810
ScalarMix+SelfAttn	1.213	0.740
MeanLayer+SelfAttn	1.123	0.781
MaxLayer+SelfAttn	1.093	0.797
MeanToken	1.050	0.812
ScalarMix+MeanToken	1.072	0.805
MeanLayer+MeanToken	1.061	0.810
MaxLayer+MeanToken	1.052	0.812

(b) Closed track: German

Model	RMSE	ρ
baseline	1.126	0.802
SelfAttn	1.120	0.805
ScalarMix+SelfAttn	1.325	0.715
MeanLayer+SelfAttn	1.404	0.675
MaxLayer+SelfAttn	1.177	0.782
MeanToken	1.112	0.801
ScalarMix+MeanToken	1.138	0.798
MeanLayer+MeanToken	1.117	0.805
MaxLayer+MeanToken	1.115	0.807

(c) Closed track: Spanish

Table 6: Shared task performance and hyperparameter results of submitted models. Models were fine-tuned on the train set and evaluated on the development set.

Configuration	Track	L1	Learning rate	Weight decay	Warmup step	RMSE	ρ
MTL+SelfAttn	open	CN	4e-05	0.0	100	0.885	0.851
MTL+SelfAttn	open	DE	4e-05	0.0	100	1.053	0.829
MTL+SelfAttn	open	ES	4e-05	0.0	100	1.002	0.830
MTL+MeanToken	open	CN	4e-05	0.1	200	0.895	0.847
MTL+MeanToken	open	DE	4e-05	0.1	200	1.056	0.829
MTL+MeanToken	open	ES	4e-05	0.1	200	0.990	0.834
MTL+MeanLayer+MeanToken	open	CN	5e-05	0.1	200	0.899	0.845
MTL+MeanLayer+MeanToken	open	DE	5e-05	0.1	200	1.061	0.826
MTL+MeanLayer+MeanToken	open	ES	5e-05	0.1	200	1.009	0.827
MTL+MaxLayer+MeanToken	closed	CN	4e-05	0.1	100	0.905	0.843
MTL+MeanToken	closed	CN	5e-05	0.0	100	0.932	0.832
MTL+SelfAttn	closed	CN	5e-05	0.0	100	1.185	0.710
MTL+MeanLayer+MeanToken	closed	DE	5e-05	0.0	200	1.011	0.827
MTL+MaxLayer+MeanToken	closed	DE	5e-05	0.0	200	1.032	0.818
MTL	closed	DE	4e-05	0.0	100	1.039	0.815
MTL	closed	ES	5e-05	0.1	200	1.084	0.818
MTL+MeanToken	closed	ES	4e-05	0.1	200	1.114	0.807
MTL+SelfAttn	closed	ES	3e-05	0.0	200	1.142	0.796

Search space for hyperparameters: learning rate (1e-5, 2e-5, 3e-5, 4e-5, 5e-5), weight decay (0, 0.1), warmup step (100, 200).

Glite at BEA 2026 Shared Task 1: Holistic Difficulty Models Dominate, Feature Engineering Closes the Gap in L1-Aware Vocabulary Difficulty Prediction

Vassili Philippov¹, Dmitrii Andreev¹, Pavel Katunin¹, Anton Nikolaev²

¹Glite, {vassili, dmitrii, pavel}@glite.ai

²School of Biosciences, The University of Sheffield, Sheffield, UK
a.nikolaev@sheffield.ac.uk

Abstract

This paper describes our submission to the BEA 2026 Shared Task on L1-Aware English Vocabulary Difficulty Prediction. We build per-L1 CatBoost regressors over 1,161 candidate linguistic, psycholinguistic, dictionary, and LLM-derived features drawn from 129 feature sets; out-of-fold predictions from fine-tuned encoder and decoder-LLM regression heads enter the model as additional features. Features are selected via Recursive Feature Elimination with nested cross-validation, producing compact per-L1 models of 29–150 features per run. For the closed track we introduce a per-feature-column compliance audit that classifies 57 of 129 feature sets as track-eligible under the organiser rulings, an audit that forced a rebuild of the selection and ensembling pipelines in the final week. We further show that decoder-LLM LoRA regression heads — LLaMA-3.1-8B being the single strongest model in our pool — provide the largest marginal gains in the open track, and that a simpler per-L1 CatBoost on RFE-selected features matches or exceeds Ridge-stacking ensembles over the same base models. Our systems ranked 1st in the closed track and 2nd in the open track on all three L1s (Spanish, German, Mandarin), reducing baseline RMSE by 29.9% in the closed track and 35.9% in the open track on average.

1 Introduction

Vocabulary knowledge is a cornerstone of L2 language proficiency and a primary determinant of reading comprehension, listening, and writing ability. Reliable per-item difficulty calibration is the bottleneck for adaptive language assessment, CEFR-aligned curriculum design, learner-knowledge prediction, and personalised learning. Traditional calibration requires costly pretesting against representative learner populations; data-driven NLP methods offer a scalable alternative, provided they can be shown to recover the psychometric difficulty that learners actually experience.

The BEA 2026 Shared Task introduces two features absent from earlier shared tasks on word difficulty. First, *L1-awareness*: items are presented with a localised prompt in the learner’s first language, and the task is to predict difficulty for Spanish, German, and Mandarin Chinese L1 speakers separately. Second, labels are *psychometrically calibrated*: the target is a GLMM-based difficulty estimate drawn from the British Council’s Knowledge-based Vocabulary Lists (Schmitt et al., 2024), fit over roughly 3.3M test responses from more than 100,000 learners. Unlike most public vocabulary benchmarks, the task measures *productive* knowledge with an exact-spelling requirement: the learner must produce the target from a partial-spelling clue (first letter + word length), an L1 translation, and a localised context — a setup that materially shifts what difficulty is being measured. Prior work addressed related problems — CWI (Paetzold and Specia, 2016; Yimam et al., 2018), LCP (Shardlow et al., 2021), and MLSP (Shardlow et al., 2024) — without learner-specific context or psychometric targets. The task’s closed/open split — closed (no LLMs, no extra data) vs. open (unrestricted) — adapts the Restricted/Unrestricted distinction used in the BEA 2019 grammatical-error-correction shared task (Bryant et al., 2019), and isolates the contribution of resource access.

We describe a system that places first in the closed track and second in the open track across all three L1s. The contributions are:

- **A closed-track system that places 1st on all three L1s**, reducing baseline RMSE by 28.2–31.9% (average 29.9%).
- **An open-track system placing 2nd on all three L1s**, with an average baseline reduction of 35.9%.
- **A per-feature-column closed-track compliance framework** applied to 129 feature sets,

classifying 57 as eligible and 72 as ineligible under the organiser rulings — a 56% rejection rate that required rebuilding selection and ensembling pipelines in the final week of the competition.

- **Empirical evidence that end-to-end holistic predictors are the dominant signal.** A leave-one-family-out ablation shows removing the holistic-models family costs +0.102 RMSE (an order of magnitude more than any other), driven by decoder-LLM LoRA regression heads — LLaMA-3.1-8B alone attains 0.831 at 0.13% trainable parameters. The holistic family is considerably larger than comparator families (68 vs 5–14 sets), which amplifies the ablation effect; per-set performance in Table 7 confirms the gap is not due to size alone. The second-largest lever is the *virtual-learner* family (+0.006 RMSE): weak LLMs used as proxy learners rather than difficulty judges.
- **A candid post-mortem of a data-leakage incident** in which four feature sets leaked the target variable into LLM prompts or cross-fold predictions; the leaked ensemble scored RMSE 0.609 before invalidation, and was caught by structural verification of every feature artefact.
- **An autonomous research framework** (§4) that evaluated more than 1,000 candidate features across 270+ tracked experiments, with specified artefacts and an immutable audit trail.

Section 2 reviews prior shared tasks; Section 3 describes the data; Section 4 describes the system; Section 5 details the experiments; Section 6 presents results; and Section 7 analyses per-L1 errors, what worked, and what did not.

2 Related Work

Shared tasks on word difficulty. Complex Word Identification framed difficulty as binary classification for L2 readers (Paetzold and Specia, 2016; Yimam et al., 2018). Lexical Complexity Prediction moved to continuous regression with multiple annotators per item (Shardlow et al., 2021), and the Multilingual Lexical Simplification Pipeline extended the setup to multiple languages and the downstream simplification task (Shardlow et al., 2024). None of these tasks conditions prediction on the learner’s L1 or uses psychometrically calibrated targets; the BEA 2026 task (Felice and Skidmore, 2026; Schmitt et al., 2024) is, to our knowledge, the first shared task to combine both.

Track structure. Our closed/open split follows the Restricted vs. Unrestricted distinction introduced by the BEA 2019 grammatical-error-correction shared task (Bryant et al., 2019), which also defined a third Low-Resource track. BEA 2019 enforced track compliance through submitted system descriptions; our closed-track compliance framework (§4.4) is a finer-grained realisation that audits eligibility at the level of individual feature columns.

Methodology precedents. Gombert et al. (2024) won the BEA 2024 Item Difficulty shared task with a transformer-encoder regression that uses scalar mixing across encoder layers, multi-task training on item difficulty and response time, and rational-network regression heads. Several other BEA 2024 systems stacked multiple encoder regressors via linear meta-learners. We combine elements of both patterns while diverging in three respects: (i) the pool is enlarged to include decoder-LLM LoRA regression heads alongside encoder regressions; (ii) per-L1 gradient boosting replaces the linear meta-learner as the fusion layer; and (iii) features are selected by Recursive Feature Elimination with nested cross-validation rather than all-in stacking, which reduces dimensionality before the final fit and limits overfitting on the relatively small (~16k-row) out-of-fold validation set.

External resources. Individual feature families rely on established psycholinguistic norms and L2-oriented vocabulary lists, including concreteness and prevalence norms (Brysbaert et al., 2014, 2019), the SCOPE psycholinguistic metabase (Gao et al., 2023), the CEFR-J wordlist (Tono, 2019), CEFR-annotated WordNet (Kikuchi et al., 2025), Chinese–English L2 AoA norms (Wang and Chen, 2020), the MorphoLex morphological database (Sánchez-Gutiérrez et al., 2018), the LADEC database of English compounds (Gagné et al., 2019), the language-specific SUBTLEX subtitle frequency corpora for Chinese, German, and Spanish (Cai and Brysbaert, 2010; Brysbaert et al., 2011; Cuetos et al., 2011), and the OpenSubtitles parallel corpus (Lison et al., 2018). Our open-track submission additionally uses a proprietary sense-level English lexical resource with per-sense difficulty estimates. We cite all of these in-line in §4.2.

Split	# items / L1	# L1s	# rows
Train	6,091	3	18,273
Dev	677	3	2,031
Test	748	3	2,244
Total	7,516	3	22,548

Table 1: Dataset statistics. Item ids are shared across L1s, so the same English word appears in all three L1 views. Train, dev, and test splits are disjoint by item id (ranges 1–6091, 6092–6768, and 6769–7516 respectively).

3 Task and Data

Task. Given an English target word presented to a learner whose L1 is Spanish (ES), German (DE), or Mandarin Chinese (CN), together with a localised prompt (a partial-spelling clue, a translation, and a context sentence in the L1), predict the word’s psychometric difficulty as a continuous GLMM score (lower score = more difficult). A canonical Spanish example presents the cue “*h_____*” alongside a Spanish context sentence and the translation *casa*, with the target English word being *house*. Responses are graded correct only when the spelling matches exactly, so orthographically irregular forms carry a spelling-error component in the GLMM difficulty alongside the lexical-knowledge component.

Data. The task distributes item-level data for 7,516 English target words, each presented in three L1 variants. Table 1 summarises the splits. The labels are derived from a GLMM fit over approximately 3.3M real learner responses from more than 100,000 British Council KVL test-takers (Schmitt et al., 2024). Test labels were withheld during the shared-task evaluation period and have since been released publicly on the shared-task GitHub repository.¹

Metrics. The shared task ranks systems by Root Mean Squared Error (RMSE, lower is better) averaged across L1s, with Pearson correlation reported as a secondary metric. We use RMSE as the primary decision metric throughout: all feature selection, ensemble pruning, and submission choices are made on K-fold RMSE.

Tracks. The *closed* track forbids large language models and additional training data beyond the shared task release, while allowing publicly avail-

able off-the-shelf pretrained transformers and linguistic databases. The *open* track places no restrictions. A team may submit up to three runs per track per L1.

4 System

4.1 Overview

Our submission is a per-L1 gradient-boosted regressor consuming a broad pool of heterogeneous features, including the out-of-fold (OOF) predictions of fine-tuned encoder and decoder-LLM regression heads. Figure 1 gives the end-to-end picture. The feature pool comprises 129 feature sets totalling 1,161 numeric columns, organised by seven domain families plus a small derived bucket (§4.2). Per L1, features are reduced with Recursive Feature Elimination under nested cross-validation, yielding compact sets of 29–150 features that train a single CatBoost regressor (Prokhorenkova et al., 2018) on the union of train and dev.

Autonomous research framework. Our submission was developed inside the Glite Autonomous Research Framework.² The framework enforces task isolation (one directory, one branch, one pull request per experiment), artefact specifications checked by verifiers before commit, and an immutable log of every experiment step. This enabled us to evaluate over 1,000 candidate features across 270+ tracked experiments, with automatic metric aggregation and append-only audit trails. The data-leakage incident described in §7.3 was detected precisely because every feature CSV is spec-verified and every fold-level score is traceable to its code revision.

4.2 Features

We organise our 129 feature sets around *what the feature measures about item difficulty* rather than how it was produced. Seven domain families — plus a small derived bucket — follow. Appendix B has per-family counts and the top-ranked sets per family (Table 7).

Target-word lexical properties (14 sets). Surface form, frequency, morphology, and semantic-lexical properties of the English target word itself: word length, corpus log-frequency, morphological decomposition (Sánchez-Gutiérrez et al., 2018), compound transparency (Gagné et al., 2019), neighbourhood density, concreteness (Brysbaert et al.,

¹<https://github.com/britishcouncil/bea2026st>

²<https://github.com/GliteTech/glite-arf>

129 feature sets → 1,161 numeric feature columns, grouped by seven domain families

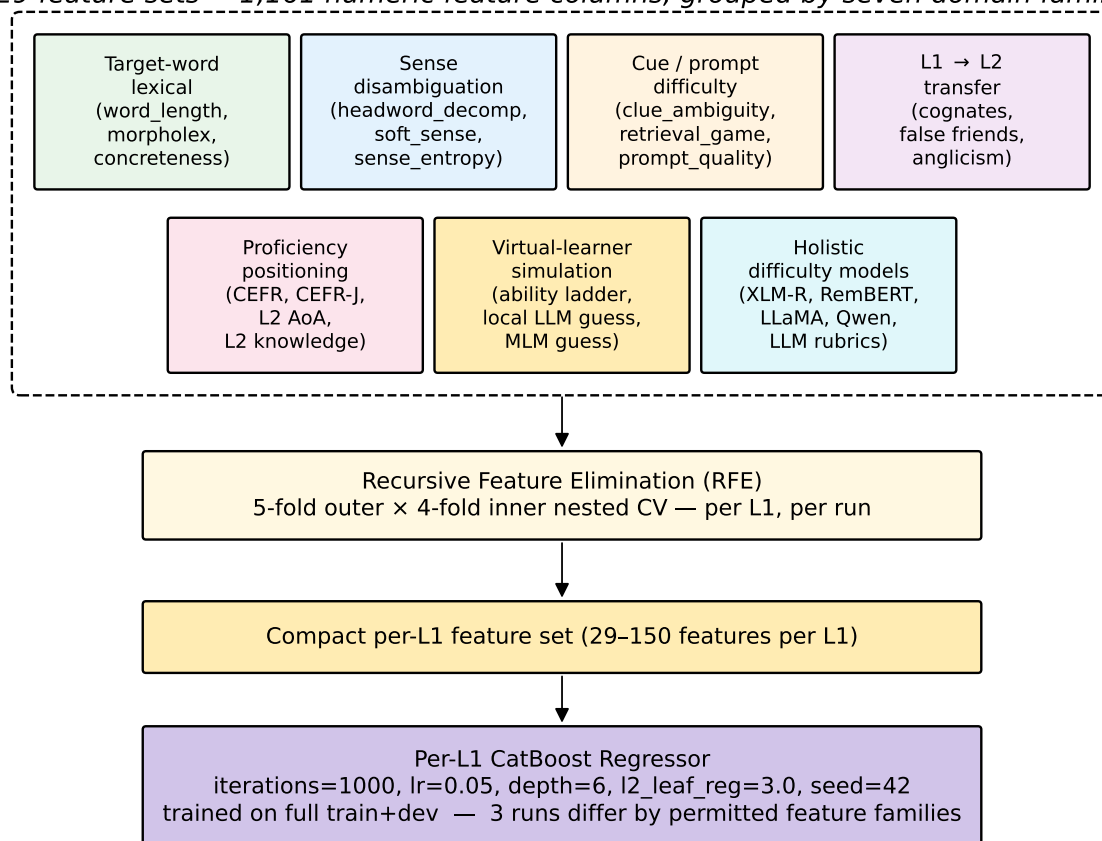


Figure 1: System architecture. Seven feature families feed into per-L1 RFE with nested CV, which yields compact per-L1 feature sets consumed by CatBoost. Transformer and decoder-LLM signals enter as out-of-fold predictions rather than via a separate stacking layer. Three submission runs differ in which feature sets and models are permitted.

2014), prevalence (Brysbaert et al., 2019), the SCOPE psycholinguistic metabase (Gao et al., 2023), and lemma-level features from a proprietary sense-level English lexical resource. These are the features a lexicographer would call out first when asked why a word is hard.

Sense-level disambiguation (7 sets). For polysemous words the tested sense is often not the dominant sense, so features that model *which* sense is tested matter. Each KVL item is first mapped to a sense in a proprietary English dictionary via an LLM word-sense-disambiguation prompt given the target, clue sentence, and L1 translation. The family then covers sense-level dictionary features (212 columns; solo Pearson 0.704), morphological headword decomposition, soft sense distributions, sense entropy, and a polysemy-contrast LLM feature.

Cue / prompt difficulty (6 sets). Features that analyse the *clue* (partial spelling + L1 translation

+ L1 context), not the target word itself: clue-ambiguity (dictionary words that match the same first-letter + word-length pattern), a retrieval-game information-theoretic model of candidate competition, semantic-competition counts of viable alternatives, LLM-rated prompt quality, prompt-framing variants, and a context-only guess feature. These capture how tightly the cue constrains the answer.

L1 → L2 transfer (14 sets). Cross-linguistic features about whether the learner’s L1 helps or hurts: cognate distance, CogNet-detected cognates, a Spanish expert cognate list, anglicism detection, false-friend flags, LLM language-similarity ratings, phonological transfer, cross-lingual orthographic neighbourhood, Chinese script-complexity, L1-side frequency (corpus and subtitle), etymology, translation ambiguity, and LLM back-translation quality.

L2 proficiency positioning (8 sets). Empirical evidence of where the target word sits on

the learner- developmental scale: CEFR levels from EFLLex and Words-CEFR (Tono, 2019), the CEFR-J / JACET Asian-learner wordlists, sense-level CEFR from CEFR-annotated WordNet (Kikuchi et al., 2025), Brysbaert’s L2 yes/no word-knowledge norms, L2 age of acquisition (Wang and Chen, 2020), empirical learner production rates, and Cambridge cloze-exam presence. Every set in this family is closed-track eligible.

Virtual-learner simulation (7 sets). A distinctive family in our system: features produced by letting a weaker or ability-graded model *attempt to answer* the item and recording whether it succeeds, its confidence, or its surprisal. The LLM here acts as a proxy learner, not as a difficulty judge. This family includes an ability ladder over four OpenAI models of graded size, guesses from four local small models, LLM clue- and description-based guessing, LLM answer entropy, token-surprisal features, and masked-LM pseudo-likelihood guessing. All are open-track-only except the masked-LM variant (closed-eligible).

Holistic difficulty models (68 sets). End-to-end predictors that read the full item and emit a scalar difficulty estimate, entering the pipeline as out-of-fold features. Covers fine-tuned encoders — XLM-R (Conneau et al., 2020), RemBERT (Chung et al., 2021), ELECTRA-base (Clark et al., 2020), and mBERT per L1 with TAPT (Gururangan et al., 2020) and R-Drop (Liang et al., 2021) under multiple seeds; scaled encoders (XLM-R-XL 3.5B, XLM-R-XXL 10.7B) trained with LoRA (Hu et al., 2022); decoder-LLM LoRA regression heads for LLaMA-3.1-8B (Dubey et al., 2024), Qwen-2.5-7B (Yang et al., 2024), and Mistral-7B (Jiang et al., 2023); LLM rubric variants; and LLM embeddings. LoRA uses rank 16, $\alpha = 16$, targeting $q_proj/k_proj/v_proj$ (0.13% trainable) with a scalar regression head on the last-token hidden state. LLaMA-3.1-8B achieves the single strongest K-fold RMSE (0.831), beating the 10.7B XLM-R-XXL encoder (0.966) despite fewer parameters; Mistral-7B scores 0.951 on average but suffers a fold-4 collapse (§7.3).

Derived / second-order (5 sets). Features whose inputs are other features’ outputs: pairwise interactions, model-disagreement metrics, per-item CatBoost feature-group ablation sensitivities, and a Ridge distillation from the SemEval-2021 LCP dataset.

4.3 Feature selection and final model

Selection. For each L1 and each submission run, we run Recursive Feature Elimination with nested cross-validation (5-fold outer, 4-fold inner). The outer folds partition item ids; inner folds select the smallest feature set whose K-fold RMSE is within 0.001 of the optimum. The resulting per-L1 feature counts are summarised in Appendix A.

Final model. For each (track, run, L1) combination we train a single CatBoost regressor with `iterations=1000`, `learning_rate=0.05`, `depth=6`, `l2_leaf_reg=3.0`, and `seed=42` on the union of the train and dev splits. No early stopping is used. The CatBoost feature matrix contains the RFE-selected subset of linguistic, psycholinguistic, and dictionary features, plus the OOF predictions of the fine-tuned transformer and decoder-LLM regression heads treated as ordinary numeric features.

Architecture alternatives considered. A 10-model Ridge meta-learner over three global CatBoost variants plus seven transformer OOF columns (per-L1 $\alpha = 500, 500, 100$) scored K-fold RMSE 0.745 versus 0.740 for the CatBoost pipeline. Its coefficients corroborated two observations used later in §7.3: gradient-boosted features dominate ($\approx 88\%$ of total weight), and Mistral-7B receives near-zero weight. We submitted the simpler CatBoost pipeline; a greedy pruning over the 63-model Ridge pool (Figure 2) further confirms that six models match the full ensemble.

Three submission runs per track. The three runs differ in which feature sets and models are permitted at RFE time. Run 1 is the full RFE-optimal pipeline. Run 2 is a resource-robust variant: for the open track it excludes the proprietary dictionary; for the closed track it excludes the large XLM-R-XL encoder. Run 3 is the most conservative: transformers-only for the open track, and a larger feature-rich RFE set for the closed track. Appendix A lists per-run feature counts and K-fold RMSE.

4.4 Closed-track compliance engineering

The closed track prohibits generative LLMs, paid APIs, additional training data, and cross-L1 training, while allowing publicly available off-the-shelf pretrained transformer encoders, standard NLP tools, and public linguistic databases (WordNet, CEFR wordlists, SUBTLEX, EFLLex, MorphoLex,

LADEC, SCOPE, etc.). Feature selection and final CatBoost training were performed separately for each L1, using only that L1’s provided train and dev rows. Because features are generated from heterogeneous code paths and our feature pool is large, we developed a per-column compliance schema that records, for each feature column, which external models, APIs, datasets, and computation paths were used to produce it. Every feature set is audited against five organiser rulings issued in March 2026: (1) difficulty-in-prompt leakage, (2) LLMs predicting the target word, (3) paid embedding APIs, (4) LLMs estimating learner production probability, and (5) LLMs used purely for rubric scoring without the target. Seventeen features were reclassified after these rulings, and two further reclassifications came from our internal audit (inadvertent training-data contamination and external-data usage). Of 129 feature sets (1,161 columns), 57 (249 columns) remained closed-track-eligible after the audit — a 56% rejection rate that forced a full rebuild of the RFE and submission pipelines in the final week. A 469-line closed-track LLM-usage disclosure submitted with our system enumerates all 14 pretrained models used in the closed-track submission; all are encoder-only or encoder-decoder, with no generative LLMs and no paid APIs. We view this per-column audit as a generalisable methodological contribution for any shared task with resource restrictions.

5 Experiments

Folds. We use a single $K=5$ fold assignment over item ids, shared across L1s, so that any English word appears in the same fold in all three L1 views. This prevents cross-L1 leakage via joint training. The fold assignments are persisted deterministically and reused across every experiment in the project.

Hyperparameters. CatBoost uses `iterations=1000`, `lr=0.05`, `depth=6`, `l2_leaf_reg=3.0`, `seed=42`. Encoder regressors (XLM-R, RemBERT, ELECTRA, mBERT) use batch size 16 (or 4 for XL/XXL), max length 128, AdamW with linear warmup and decay, 3 epochs for per-L1 variants with TAPT+R-Drop ($\lambda = 1.0$), and seed sweep over $\{123, 456, 789\}$. LoRA adapters target `q_proj/k_proj/v_proj` with rank 16, $\alpha = 16$, dropout 0.1; decoder-LLM regression heads read the last-token hidden state through an MLP of width 2048 with GELU. Full

hyperparameters are listed in Appendix C.

Hardware and compute. CatBoost models and small-encoder fine-tuning ran locally on Mac Silicon (48 GB unified memory); scaled encoders (XLM-R-XL 3.5B, XLM-R-XXL 10.7B) and the 7–8B decoder-LLM regression heads (LLaMA-3.1-8B, Qwen-2.5-7B, Mistral-7B) were trained on rented A100-class GPUs. Total compute was approximately 100 wall-hours and roughly \$500 in commercial LLM API spend.

6 Results

Main result. Our system placed 1st in the closed track on all three L1s and 2nd in the open track on all three L1s (Table 2). The average RMSE reduction over the official baseline is 29.9% closed (28.2% ES, 29.6% DE, 31.9% CN) and 35.9% open (37.1% ES, 34.5% DE, 36.2% CN).

Ablation by domain family (Table 3). Table 3 reports two complementary analyses. *Leave-one-out* (left block) removes one family at a time and re-runs RFE; *family alone* (right block) keeps only that family and runs RFE over it. Together these measure *necessity* and *sufficiency*. *Holistic difficulty models* are both: removing the family costs +0.102 RMSE (an order of magnitude more than any other), and alone the family attains K-fold RMSE 0.772 / Pearson 0.900 with just 29 features — within 0.03 of the full system. *Virtual-learner simulation* is the clearest second-tier contributor: +0.006 RMSE to remove and Pearson 0.743 alone. The other five families show a striking asymmetry: their leave-one-out costs are small (+0.001 to +0.003), yet alone they reach Pearson 0.68–0.78 — classical-tabular signal that is largely absorbed by the holistic models when present but remains substantive when isolated. *Cue / prompt difficulty* is the strongest standalone classical family (r 0.782).

Top features per L1. The top-15 RFE features per L1 (Appendix D, Table 10) reveal a universal core: all four LLaMA-3.1-8B seeds, all four Qwen-2.5-7B seeds, the ability-ladder mean success rate, the LLM-based counterfactual cue-sensitivity score, and the learner’s L2 vocabulary-knowledge rank — all appear in the top 15 for every L1. L1-specific features surface in the mid-ranks: surprisal and interaction features for ES, an additional XLM-R-XL variant for DE, and a frequency-based cue-sensitivity variant plus a closed-eligible per-L1 XLM-R-large-TAPT model for CN.

Track	System	ES		DE		CN		Avg
		RMSE	r	RMSE	r	RMSE	r	RMSE
Closed	1. Ours	0.903	0.877	0.885	0.871	0.776	0.889	0.855
Closed	2. Best other team [†]	0.975	0.858	0.903	0.869	0.816	0.874	0.898
Closed	3. Best other team [‡]	0.976	0.857	0.963	0.844	0.820	0.879	0.920
Closed	Baseline	1.257	0.765	1.258	0.773	1.140	0.753	1.218
Open	1. Sakura	0.742	0.919	0.723	0.916	0.630	0.928	0.698
Open	2. Ours	0.754	0.916	0.764	0.905	0.660	0.920	0.726
Open	3. TeamXBC	0.876	0.885	0.826	0.888	0.722	0.904	0.808
Open	Baseline	1.198	0.783	1.166	0.786	1.034	0.804	1.133

Table 2: Official test-set leaderboard excerpt, summarised at team level using each team’s best submitted run per L1. RMSE is the primary ranking metric (lower is better); r is Pearson. **Bold** marks the best per-column entry for that track. Closed track had 18–19 unique participating teams per L1; open track had 6–7. [†]The 2nd-best closed-track team is *uogal* for ES and DE, and *Sakura* for CN. [‡]The 3rd-best closed-track team is *AIDA* for ES, *Sakura* for DE, and *uogal* for CN.

Configuration	Leave-one-out		Family alone		
	K-fold RMSE	Δ RMSE	#feats	K-fold RMSE	Pearson
Full (all 7 families)	0.740	—	—	—	—
– Holistic difficulty models	0.842	+0.102	29	0.772	0.900
– Virtual-learner simulation	0.746	+0.006	75	1.194	0.743
– L2 proficiency positioning	0.743	+0.003	34	1.286	0.684
– Cue / prompt difficulty	0.742	+0.002	47	1.105	0.782
– Sense disambiguation	0.742	+0.002	62	1.233	0.718
– L1 $\rightarrow$ L2 transfer	0.741	+0.001	54	1.261	0.696
– Target-word lexical	0.741	+0.001	121	1.276	0.693

Table 3: Open-track ablation by domain family (§4.2). Left block (*leave-one-out*) deletes the named family and re-runs RFE over all remaining features, reporting the rebuilt system’s K-fold RMSE and its gap to the full system (full system: 66 RFE-selected features, K-fold RMSE 0.740). Right block (*family alone*) keeps only the named family and runs RFE over its features, reporting the optimal feature count, K-fold RMSE, and Pearson of the family-only system. Same folds, hyperparameters, and RFE procedure throughout. Rows sorted by Δ RMSE (most damaging removal first).

K-fold $\rightarrow$ test calibration. Sixteen of 18 submitted runs scored *better* test RMSE than K-fold RMSE (mean $\Delta = -0.022$ closed, -0.012 open); test Pearson exceeded K-fold Pearson on every comparable run (Appendix B). The two runs where test was worse than K-fold have small magnitude ($+0.001$ and $+0.015$ RMSE). We therefore find no empirical evidence of substantial K-fold overfitting across our 1,000+ feature search, and this pattern is consistent with the leak audit described in §7.3, since a residual leak would inflate K-fold relative to test.

Greedy ensemble pruning. For the 10-model Ridge architecture considered in §4.3 we independently ran greedy forward selection over a larger 63-model Ridge pool (Appendix A, Figure 2). The best single model achieves 0.814 (a CatBoost RFE variant). Adding the second (LLaMA-3.1-8B) drops RMSE by 0.056. Six models suffice to match the full 63-model ensemble (0.745 vs 0.745). The

optimum is reached at 39 models (0.739). Beyond that, the next 24 additions are neutral or harmful, and the 63rd model added — *mistral_7b* — is the single worst addition, raising RMSE by 0.004.

Gap to first place in the open track. Sakura placed first on all three open-track L1s with test RMSE 0.742 / 0.723 / 0.630 (ES / DE / CN). Our second-place gaps are small on Spanish ($+0.012$) but widen on German ($+0.041$) and Mandarin ($+0.030$). We suspect Sakura’s fine-tuned-LLM pipeline captures L1-specific morphology — Germanic compounding in particular — that our classical tabular features under-represent; closing these gaps is a concrete direction for future work.

Closed-track margin is wider. Our closed-track first-place margins over the second-best team (*uogal* for ES/DE, *Sakura* for CN) are 0.072 / 0.018 / 0.040 RMSE — roughly $3\times$ our open-track gap to Sakura on ES and CN, and about $4\times$ on DE.

This asymmetry is consistent with the closed-track restrictions penalising systems whose open-track advantage depends on paid LLM APIs: once generative LLMs are removed, the per-column compliance audit (§4.4) and our 57-set eligible pool produce a feature base competitive enough to lead each L1.

7 Analysis

7.1 Per-L1 error patterns

The three L1s show systematically different residual structure. Spanish items with Latinate roots are under-predicted as easier than they are; items with Germanic roots or noun morphology shared with French cognates are the opposite. German residuals are reduced most by the length-ratio feature and by the XLM-R-XL OOF column, which carries German-sensitive morphology signal absent from the 7B decoder-LLMs. Mandarin has the lowest RMSE on all systems — the CN target distribution is narrower — and benefits least from cognate-style features, instead drawing on the character-level Chinese script-complexity feature and per-L1 RemBERT OOF. All three L1s exhibit regression-to-mean compression of roughly 20% (slope ≈ 0.80) that shrank between ensemble versions but did not vanish (quantified further in the Limitations section).

7.2 What worked

Holistic predictors are necessary and nearly sufficient. Seven of the top-10 RFE-ranked features for every L1 are decoder-LLM OOF columns. A single LLaMA-3.1-8B head (0.831 K-fold RMSE) beats XLM-R-XXL (0.966, 42% more parameters) at 0.13% trainable parameters; adding Qwen-2.5-7B drops RMSE a further 0.010. The ceiling analysis in Table 3 sharpens this: an RFE pass restricted to the 68 holistic sets converges to 29 features at K-fold RMSE 0.772 / Pearson 0.900 — within 0.03 RMSE of the full system. In greedy forward selection LLaMA is the second model added (-0.056 RMSE) and Qwen the third (-0.010).

Virtual-learner simulation is a distinct signal. Removing the *virtual-learner* family costs $+0.006$ RMSE (Table 3), larger than any classical-tabular or cue/sense family; alone the family reaches Pearson 0.743. Letting a weak LLM *attempt* the item therefore adds signal the decoder-LLM regression heads do not already absorb — a

different way of using LLMs from the dominant regression-head paradigm.

Simpler ensemble beat Ridge stacking. A single per-L1 CatBoost consuming transformer and LLM OOF predictions as plain features matched or exceeded every Ridge-stacking alternative; greedy pruning shows only 6 of 63 models suffice to saturate the Ridge ensemble.

Column-level compliance preserved signal. Operating the March 2026 compliance audit at the column level rather than the feature-set level preserved 249 columns across 57 closed-eligible sets that a set-level audit would have discarded, including the per-L1 RemBERT and XLM-R-large-TAPT OOF columns that dominate closed-track feature importance.

7.3 What did not work

Data-leakage post-mortem. Our ensemble v13 reached K-fold RMSE 0.609 — implausibly strong. Four leaking feature sets were identified: two passed the GLMM target into LLM prompts asking the model to judge annotation quality; a third used cross-L1 out-of-fold predictions with an item-id split shared across L1s; a fourth produced residual-prediction features circular with the training loss. Quarantining all four and rebuilding every downstream artefact yielded a clean v14 at 0.802. We cannot rule out undetected remaining leaks, but the K-fold $\rightarrow$ test calibration (§6, Appendix B) is consistent with the audit: test RMSE beats K-fold on 16 of 18 runs, whereas a residual leak would invert that pattern.

Mistral-7B is actively harmful in the ensemble. Of three decoder LLMs trained with the same LoRA recipe, LLaMA (0.831) and Qwen (0.840) were useful, but Mistral-7B (0.951) was not. Its average hides a fold-4 collapse to 1.36 (folds 0–3: 0.83–0.89), suggesting an initialisation pathology rather than a data issue. In greedy forward selection over the 63-model Ridge pool Mistral is the final addition and adds $+0.004$ RMSE — the single largest harmful contribution — and a parallel 10-model Ridge independently assigns it near-zero weight. The submitted system excludes Mistral.

Seven recurring failure patterns. Across 37 negative experiments we observed seven recurring patterns: (i) transformer architectural surgery (scalar-mix, multi-task, CharCNN) made no difference over a tuned XLM-R-base; (ii) adversarial

training did not help at 20k rows; (iii) tabular features saturated at ~ 20 sets; (iv) derived features (SHAP, counterfactual sensitivity, residual regressors) tended to be circular with the target; (v) only 3 of 13 psycholinguistic databases added marginal signal; (vi) HPO and bagging moved Pearson by ≤ 0.001 per 100 trials; (vii) Pinyin/script normalisation destroyed information XLM-R’s tokeniser handles natively.

8 Conclusion

Our system placed 1st closed / 2nd open on all three BEA 2026 L1s. The architecture is a per-L1 CatBoost on RFE-selected features plus OOF predictions from fine-tuned transformer and decoder-LLM regression heads. Three findings stand out: decoder-LLM LoRA heads dominate open-track importance (LLaMA-3.1-8B strongest); column-level compliance preserved 249 closed-eligible columns a set-level audit would have rejected; and six models match a 63-model Ridge ensemble, with Mistral-7B actively degrading it. Future work: distil the decoder-LLM signal into closed-track-legal form, and extend to more L1s.

Limitations

Our open-track system relies on a proprietary sense-level English lexical resource with difficulty estimates; reproducing the exact open-track numbers requires licensing of this resource. The closed-track system uses only public resources and can be reimplemented from the feature descriptions and hyperparameters given in §4 and Appendix C. We evaluated only three L1s — Spanish, German, and Mandarin Chinese — all drawn from the British Council KVL test-taker population; generalisation to typologically distant L1s such as Arabic or Japanese, and to learner populations outside paid-test-taker demographics, is untested. Our feature search explored more than 1,000 candidate features using autonomous agents under human supervision; while every artefact is spec-verified and every cross-validation fold is deterministic, the scale of the search creates an *a priori* risk of overfitting to the K-fold protocol. In practice this risk did not materialise: 16 of 18 submitted runs scored better on the held-out test set than on K-fold (mean Δ RMSE = -0.017), and test Pearson exceeded K-fold Pearson on every comparable run (Appendix B) — indicating that nested cross-validation plus dev-set diagnostics were sufficient

for calibration at this search scale. Our best ensembles exhibit roughly 20% regression-to-mean compression (slope ≈ 0.80): the hardest items are systematically under-predicted by -0.55 GLMM units and the easiest over-predicted by $+0.43$; this pattern shrank between ensemble versions but did not disappear. Several of our strongest open-track features rely on commercial LLM APIs (approximately \$500 total across the project) and on GPU-hosted decoder-LLM regression heads requiring 55–83 GB of VRAM, available only via cloud GPU rental; despite our best submission reducing baseline RMSE by 35.9% in the open track and 29.9% in the closed track, per-item residuals of 0.6–0.9 RMSE remain, indicating the task is far from solved.

Acknowledgments

We thank the BEA 2026 Shared Task organisers — Lucy Skidmore, Mariano Felice, and Karen Dunn — for designing the task and releasing the KVL-based dataset, and the British Council for making the underlying psychometric data publicly available under a Creative Commons licence. We also thank our colleagues at Glite for supporting the development of the autonomous research framework used to coordinate the experiments reported here, and the anonymous reviewer whose comments on the submission version of this paper improved the exposition of §3 and Figure 1.

References

- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA 2019 shared task on grammatical error correction. In *Proceedings of BEA 2019*.
- Marc Brysbaert, Matthias Buchmeier, Markus Conrad, Arthur M. Jacobs, Jens Bölte, and Andrea Böhl. 2011. The word frequency effect: A review of recent developments and implications for the choice of frequency estimates in German. *Experimental Psychology*, 58(5):412–424.
- Marc Brysbaert, Paweł Mandera, Samantha F. McCormick, and Emmanuel Keuleers. 2019. Word prevalence norms for 62,000 English lemmas. *Behavior Research Methods*.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

- Qing Cai and Marc Brysbaert. 2010. SUBTLEX-CH: Chinese word frequencies based on film subtitles. *PLoS ONE*, 5(6):e10729.
- Hyung Won Chung, Thibault F  vry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *Proceedings of ICLR*.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzm  n, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.
- Fernando Cuetos, Mar  a Glez-Nosti, Analia Barb  n, and Marc Brysbaert. 2011. SUBTLEX-ESP: Spanish word frequencies based on film subtitles. *Psicol  gica*, 32:133–143.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The LLaMA 3 herd of models. *arXiv:2407.21783*.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the BEA 2026 shared task on vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Christina L. Gagn  , Thomas L. Spalding, and Daniel Schmidtke. 2019. LADEC: The large database of english compounds. *Behavior Research Methods*.
- Chuanji Gao, Svetlana V. Shinkareva, and Rutvik H. Desai. 2023. *SCOPE: The South Carolina psycholinguistic database*. *Behavior Research Methods*, 55:2853–2884.
- Sebastian Gombert, Lukas Menzel, Daniele Di Mitri, and Hendrik Drachsl  r. 2024. Predicting item difficulty and item response time with scalar-mixed transformer encoder models and rational network regression heads. In *Proceedings of BEA 2024*.
- Suchin Gururangan, Ana Marasovi  , Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.
- 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 *Proceedings of ICLR*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, and 1 others. 2023. Mistral 7B. *arXiv:2310.06825*.
- Masato Kikuchi, Masatsugu Ono, Toshioki Soga, Tetsu Tanabe, and Tadachika Ozono. 2025. CEFR-annotated WordNet: LLM-based proficiency-guided semantic database for language learning. *arXiv preprint arXiv:2510.18466*.
- Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-Drop: Regularized dropout for neural networks. In *Proceedings of NeurIPS*.
- Pierre Lison, J  rg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of LREC*.
- Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 task 11: Complex word identification. In *Proceedings of SemEval-2016*.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Proceedings of NeurIPS*.
- Claudia H. S  nchez-Guti  rrez, Hugo Mailhot, S. H  l  ne Deacon, and Maximiliano A. Wilson. 2018. MorphoLex: A derivational morphological database for english. In *Behavior Research Methods*.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based Vocabulary Lists*, volume 5 of *British Council Monographs on Modern Language Testing*. University of Toronto Press.
- Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of SemEval-2021*.
- Matthew Shardlow, Sanja   tajner, Kai North, Tharindu Ranasinghe, and Marcos Zampieri. 2024. The BEA 2024 shared task on the multilingual lexical simplification pipeline. In *Proceedings of BEA 2024*.
- Yukio Tono. 2019. CEFR-J wordlist version 1.5. Tokyo University of Foreign Studies, Tono Laboratory. <http://www.cefr-j.org/download.html>.
- Jue Wang and Baoguo Chen. 2020. A database of Chinese-English bilingual speakers: Ratings of the age of acquisition and familiarity. *Frontiers in Psychology*, 11:554785.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, and 1 others. 2024. Qwen2.5: A party of foundation models. *arXiv:2412.15115*.
- Seid Muhie Yimam, Chris Biemann, Sanja   tajner, Sian Gooding, and Matthew Shardlow. 2018. A report on the complex word identification shared task 2018. In *Proceedings of BEA 2018*.

A Submitted runs

Tables 4–5 give the per-run feature composition and per-L1 K-fold RMSE for every submitted run.

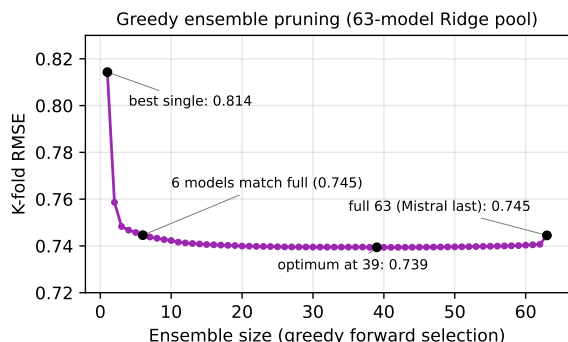


Figure 2: Greedy forward selection over a 63-model Ridge pool. Six models already match the full ensemble; the optimum is reached at 39; the last model added (Mistral-7B) is the single most harmful.

B Feature catalogue

Due to space constraints, we do not list the full 129-set catalogue here. Table 6 summarises per-family statistics, and Table 7 drills into the top feature sets in each family.

Table 7 drills into the top feature sets in each family, showing the single-set K-fold Pearson correlation and RMSE against the GLMM target. Decoder-LLM regression heads dominate, and the top closed-track-eligible signals are per-L1 encoder OOFs (RemBERT and XLM-R-large TAPT); classical linguistic and psycholinguistic features saturate around Pearson 0.55–0.59 individually, well below the transformer band.

K-fold vs test generalisation. Table 8 gives the full per-run comparison between our K-fold CV metrics (on the 5-fold item-id split of train+dev) and the official test-set metrics released by the organisers. All 18 submitted runs are listed. Sixteen of 18 runs have negative Δ RMSE (the two exceptions, open-track ES Run 3 and DE Run 1, are small in magnitude); all comparable Pearson values improve on test. K-fold Pearson is missing for Runs 2–3 where documentation recorded only RMSE.

C Hyperparameters

D Per-L1 RFE importance

Table 10 gives the top-15 RFE-ranked features per L1 for the open-track submitted system. Our

RFE pipeline records the full elimination trajectory (elimination order, per-round RMSE, optimal cut) per L1 and per track for each submitted run.

E Representative LLM prompts

We illustrate two representative prompt templates covering the two main families of LLM-derived features.

Rubric scoring (holistic LLM rater).

You are an expert L2 English teacher. Rate how difficult the target word is for an adult learner whose first language is {{L1}}, on a 0–10 scale. 0 = a near-beginner will know it; 10 = only an advanced learner will.

Target word: {{word}}
Part of speech: {{pos}}
Context sentence (English): {{clue}}
L1 translation(s): {{l1_word}}

Output only the integer score.

Ability-ladder (open track only). Four OpenAI models of graded ability are each asked to attempt the item; the feature records the fraction that recovered the target. The target word is *not* supplied; only the clue, L1 translation, and context are. Because the feature invokes generative LLMs, it is ineligible for the closed track regardless of whether the model ever sees the label.

F Compliance audit

Table 11 lists the categories of the 17 features reclassified as ineligible after the March 2026 organiser rulings. Per-column compliance metadata was maintained throughout for every feature set, including a 469-line closed-track LLM-usage disclosure submitted with our system.

G LLM API cost breakdown

Table 12 lists the ten most expensive LLM-API-based feature experiments and their positive/negative status. Total LLM spend across the project was approximately \$498. Local MLX inference (llama1b, gemma2b, phi3b, qwen1b for the local-LLM guessing feature) and GPU training were separate and not included in this table.

Run	Strategy	# features			K-fold RMSE			Avg
		ES	DE	CN	ES	DE	CN	
1	Compact (RFE-optimal)	58	49	48	0.775	0.768	0.676	0.740
2	No proprietary dict.	66	86	47	0.781	0.770	0.683	0.745
3	Transformers only	8	9	37	0.822	0.796	0.703	0.774

Table 4: Open-track runs.

Run	Strategy	# features			K-fold RMSE			Avg
		ES	DE	CN	ES	DE	CN	
1	Compact (RFE-optimal)	29	49	30	0.925	0.916	0.805	0.882
2	No XLM-R-XL	30	36	33	0.923	0.912	0.805	0.880
3	Feature-rich	50	100	150	0.926	0.916	0.814	0.885

Table 5: Closed-track runs.

Family	Sets	Cols	Closed-elig.	Solo r		Family alone	
				Best set	Mean set	K-fold RMSE	Pearson
Target-word lexical	14	149	12	0.660	0.370	1.276	0.693
Sense disambiguation	7	248	1	0.704	0.374	1.233	0.718
Cue / prompt difficulty	6	50	2	0.569	0.520	1.105	0.782
L1 $\rightarrow$ L2 transfer	14	57	9	0.442	0.307	1.261	0.696
Proficiency positioning	8	36	8	0.586	0.507	1.286	0.684
Virtual-learner simulation	7	108	1	0.576	0.458	1.194	0.743
Holistic difficulty models	68	432	24	0.884	0.752	0.772	0.900
Derived / second-order	5	81	0	0.835	0.740	—	—
Total	129	1,161	57	0.884		0.740	—

Table 6: Per-domain-family totals under the seven-family taxonomy (plus a small Derived/second-order bucket for features that operate over other families’ outputs). *Solo r* columns report the best and mean single-set K-fold Pearson within the family, taken from each set’s `feature_info.json`; *Family alone* reports an RFE pass over the family-only feature pool (Table 3 right block). The *Total* row’s *Family alone* RMSE (0.740) is the full-system K-fold result over all seven families, for reference.

Feature set	#cols	Pearson	K-fold RMSE
<i>Target-word lexical properties</i>			
Proprietary dictionary lemma-level features	82	0.660	1.330
AoA, prevalence, lexical decision	6	0.543	1.487
Corpus frequency / Zipf scale	3	0.520	1.511
SCOPE psycholinguistic metabase	18	0.505	1.530
MorphoLex morphological decomposition	8	0.400	1.623
<i>Sense-level disambiguation</i>			
Proprietary dictionary sense-level features	212	0.704	1.257
Posterior distribution over senses	7	0.547	1.481
Sense assignment uncertainty	11	0.503	1.523
Entropy over possible senses	4	0.302	1.688
Morphological decomposition of headword	5	0.219	1.730
<i>Cue / prompt difficulty</i>			
Count of viable target alternatives	4	0.569	1.456
Candidate competition from spelling clue	15	0.550	1.473
Dictionary words matching spelling pattern	7	0.537	1.641
LLM guesses from context only	10	0.493	1.541
Properties of L1 translation hint	6	0.491	1.543
<i>L1 → L2 transfer</i>			
LLM back-translation quality	7	0.442	1.588
LLM-rated L1/English similarity	5	0.440	1.566
Log-frequency of L1 source word	3	0.395	1.629
Phonological distance to English	2	0.394	1.647
Expert-annotated cognate list	8	0.381	1.615
<i>L2 proficiency positioning</i>			
Brysbaert L2 yes/no word-knowledge	5	0.586	1.434
EFLLex CEFR-level frequency trajectories	6	0.574	1.448
CEFR-J / JACET Asian-learner levels	3	0.569	1.446
CEFR level (EFLLex, Words-CEFR)	6	0.566	1.458
L2 age-of-acquisition norms	3	0.561	1.464
<i>Virtual-learner simulation</i>			
LLM token surprisal on target	10	0.576	1.447
Local small models attempt the item	24	0.574	1.455
Weak-LLM guess from description hints	10	0.565	1.505
Masked-LM guess of target from context	35	0.452	1.578
Four OpenAI models of graded ability	15	0.424	1.596
<i>Holistic difficulty models</i>			
LLaMA-3.1-8B LoRA regression head	1	0.883	0.831
Qwen-2.5-7B LoRA regression head	1	0.881	0.840
XLM-R-XXL (10.7B) LoRA + R-Drop	1	0.843	0.966
Mistral-7B LoRA regression head	1	0.833	0.951
XLM-R-large TAPT + R-Drop, per L1	1	0.821	1.062
XLM-R-XL (3.5B) LoRA + R-Drop	1	0.820	1.034
RemBERT fine-tuned per L1	–	0.814	1.024
XLM-R-XL LoRA + R-Drop, per L1	1	0.806	1.072
<i>Derived / second-order</i>			
CatBoost ablation across 6 feature groups	6	0.835	0.984
Pairwise feature interactions	40	0.827	0.993
Ensemble of LLM rubric difficulty ratings	22	0.786	1.096

Table 7: Top feature sets per domain family ranked by single-set K-fold Pearson correlation against the GLMM difficulty target. Seed variants collapsed to the base row. Families describe *what the feature measures about difficulty*: surface properties of the target word, sense disambiguation, cue / prompt difficulty, L1 transfer, proficiency positioning, virtual-learner simulation, or holistic difficulty models. Holistic models (fine-tuned transformers + LLM rubrics) dominate outright; virtual-learner probes and proficiency-positioning norms carry the next-strongest signal.

Track	L1 / Run	Approach	#feat	K-fold RMSE	Test RMSE	Δ RMSE	K-fold r	Test r	Δr
Closed	ES / 1	Compact	29	0.9250	0.920	-0.005	0.869	0.872	+0.003
Closed	ES / 2	No XLM-R-XL	30	0.9227	0.910	-0.013	-	0.876	-
Closed	ES / 3	Feature-rich	50	0.9264	0.903	-0.023	0.868	0.877	+0.009
Closed	DE / 1	Compact	49	0.9158	0.887	-0.029	0.857	0.871	+0.014
Closed	DE / 2	No XLM-R-XL	36	0.9122	0.895	-0.017	-	0.868	-
Closed	DE / 3	Feature-rich	100	0.9162	0.885	-0.031	0.857	0.871	+0.014
Closed	CN / 1	Compact	30	0.8046	0.788	-0.017	0.876	0.885	+0.009
Closed	CN / 2	No XLM-R-XL	33	0.8050	0.776	-0.029	-	0.889	-
Closed	CN / 3	Feature-rich	150	0.8136	0.785	-0.029	0.873	0.886	+0.013
Open	ES / 1	Compact	58	0.7754	0.755	-0.020	0.910	0.916	+0.006
Open	ES / 2	No prop. dict.	66	0.7813	0.754	-0.027	-	0.916	-
Open	ES / 3	Trans. only	8	0.8217	0.837	+0.015	-	0.896	-
Open	DE / 1	Compact	49	0.7678	0.769	+0.001	0.902	0.904	+0.002
Open	DE / 2	No prop. dict.	86	0.7702	0.764	-0.006	-	0.905	-
Open	DE / 3	Trans. only	9	0.7961	0.790	-0.006	-	0.898	-
Open	CN / 1	Compact	48	0.6756	0.660	-0.016	0.915	0.920	+0.005
Open	CN / 2	No prop. dict.	47	0.6830	0.662	-0.021	-	0.920	-
Open	CN / 3	Trans. only	37	0.7029	0.679	-0.024	-	0.916	-

Table 8: K-fold (internal) vs test (official) metrics for all 18 submitted runs. Negative Δ RMSE means the test score was better than K-fold; positive Δ Pearson means the same.

Model	Params	Training
CatBoost	1k trees	lr 0.05, depth 6, l2 3.0
XLM-R-base	278M	TAPT+R-Drop, bs 16, 3 epochs
XLM-R-large	550M	TAPT+R-Drop, bs 8, 3 epochs
XLM-R-XL	3.5B	LoRA r16 a16, R-Drop
XLM-R-XXL	10.7B	LoRA r16 a16, R-Drop
RemBERT	575M	per-L1, bs 16, seeds 3
ELECTRA-base	110M	per-L1, bs 16, 3 epochs
mBERT	178M	per-L1, bs 16, 3 epochs
LLaMA-3.1-8B	7.5B	LoRA r16 a16, q/k/v_proj
Qwen-2.5-7B	7.1B	LoRA r16 a16, q/k/v_proj
Mistral-7B	7.1B	LoRA r16 a16 (excluded)

Table 9: Per-family hyperparameters. All decoder LLMs use a scalar regression head on the last-token hidden state, MLP width 2048 with GELU.

#	ES	DE	CN
1	LLaMA s789	LLaMA	Qwen s789
2	Qwen s456	LLaMA s456	LLaMA s123
3	LLaMA	Qwen s123	Qwen s456
4	LLaMA s123	LLaMA s123	Qwen
5	Qwen s789	Qwen s789	Qwen s123
6	Qwen s123	XLM-R-XXL	LLaMA s789
7	al-mean	LLaMA s789	LLaMA
8	Qwen	Qwen	LLaMA s456
9	LLaMA s456	al-mean	al-mean
10	Mistral	Mistral	Mistral
11	cue-sens	cue-sens	XLM-R-lg s456
12	XLM-R-XXL	XLM-R-XL	cue-sens
13	surp-rate	Qwen s456	cue-sens-freq
14	l2-rank	l2-rank	l2-rank
15	surp-int	al-mini	al-mini

Table 10: Top-15 RFE-ranked features per L1 (open track). LLaMA and Qwen entries denote the LoRA regression OOF predictions of the corresponding seeded model; *al-mean* is the ability-ladder mean success rate; *cue-sens* is the LLM-based cue-sensitivity score; *l2-rank* is the learner’s L2 vocabulary-knowledge rank. Abbreviated to fit.

Ruling / source	# features
LLM predicts target word (Cat 2)	7
Difficulty in same API call (Cat 1)	2
Paid embedding API (Cat 3)	3
LLM estimates production prob. (Cat 5)	1
R4 contamination (internal audit)	2
Difficulty rating (internal audit)	1
External training data (internal audit)	1
Total	17

Table 11: Compliance-audit reclassifications.

Feature	Cost (USD)	Result
LLM token surprisal (3 weak models)	40.92	+
Guess from description (3 weak models)	44.34	+
LLM-rated L1/English similarity (GPT)	30.23	+
False-friend flags (GPT-4)	36.97	+
Anglicism detection (GPT-4)	34.30	+
Guess from clue (3 weak models)	63.48	+
LLM rubric (4o-mini)	1.93	best ROI
LLM rubric (reasoning)	39.15	marginal
LLM pseudo-item generation	32.80	negative
Misc. batch experiments	163.0	7/10 +
Total (approx.)	498	—

Table 12: Representative LLM API expenditure. The cheapest positive feature (\$1.93) improved RMSE by 0.015, a better return per dollar than any other.

NLP-Explorers at BEA 2026 Shared Task 1: DeBERTa–CatBoost Weighted Ensemble Approach for L1-Specific Vocabulary Difficulty Prediction

Tayyab Latif¹, Asifa Bibi¹, Sabur Butt^{2,*}, Grigori Sidorov¹, Alexander Gelbukh¹

¹Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico

²Institute for the Future of Education (IFE), Tecnológico de Monterrey, Mexico

*Corresponding author: saburb@tec.mx

Abstract

Vocabulary difficulty prediction aims to estimate how difficult a word is for a learner. This is an important problem because word difficulty is shaped not only by the word itself, but also by the learner's background and the context in which the word appears. In this work, we predict continuous difficulty scores for English target words using learner-specific information. Our approach combines a fine-tuned DeBERTa v3 Large model with a CatBoost regressor trained on transformer-based embeddings. The final score is produced through weighted ensembling, where DeBERTa provides the main prediction and CatBoost adds a smaller complementary signal. Our final system achieved RMSE scores of 1.040 for Spanish, 0.992 for German, and 0.882 for Chinese. The results were also stable across multiple runs, showing that the model behaved consistently under small changes in ensemble weight. These findings show that a simple hybrid system can provide reliable performance for vocabulary difficulty prediction. They also suggest that combining strong contextual representations with a lightweight regression model is an effective way to model learner-sensitive word difficulty.

1 Introduction

The prediction of the difficulty of a word to a learner is a significant aspect of language education, as vocabulary knowledge determines reading comprehension, assessment results, and learning materials design (Zhang and Lu, 2025). Practically, though, word difficulty is not fixed. It is not only based on the word itself, but also on the linguistic background of the learner and the context in which the word is introduced (Felice and Skidmore, 2026). This makes vocabulary difficulty prediction both educationally meaningful and technically challenging.

Consequently, there is an increasing interest in

constructing NLP models that are capable of predicting learner-sensitive difficulty scores in a consistent and scalable manner (Kapoor et al., 2026). Recent work has also started to examine vocabulary test item difficulty prediction with transformer-based models, showing that such architectures provide a strong recent baseline for this task (Skidmore et al., 2025). Even though transformer-based models have become the standard option in most NLP tasks, their usage in vocabulary difficulty prediction still raises practical concerns (Li et al., 2025). Predicting word difficulty is not only about understanding the general meaning of the word and its context. The system must capture small differences between words and project them to a continuous score that indicates learner difficulty. This complicates the task compared to many traditional text classification problems and implies that the final prediction step should be given as much attention as the transformer model itself (Li et al., 2025).

In this paper, we present our system submission to the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners (Felice and Skidmore, 2026). Our approach combines DeBERTa v3 Large and CatBoost in a hybrid prediction framework. DeBERTa serves as the main contextual model, while CatBoost uses transformer-based features to provide an auxiliary prediction signal. The final score is obtained through weighted ensembling. This design allows the system to produce a more reliable estimate of vocabulary difficulty than a transformer-only model, especially when the task depends on small lexical differences and learner-specific context.

The rest of the paper is structured in the following way. We describe the data and the evaluation setting first. Then we consider the previous work that is the most related to our work. Then, we describe our methodology, which consists of pre-processing, DeBERTa fine-tuning, the CatBoost component, and the final ensemble strategy. Then

we show the results and compare the various models and submitted runs. We end with a discussion of the broader lessons from this study and directions for future research.

2 Shared Task Data and Evaluation

The shared task focuses on vocabulary difficulty prediction for English learners. In this task, the system is given an English target word together with learner-related information, and it must predict a continuous difficulty score for that word. The data are provided separately for three learner groups based on their first language: Spanish, German, and Mandarin Chinese.

Each example includes information about the target word and the prompt shown to the learner. In our system, these fields are not treated as separate inputs. Instead, the selected text components are combined into one input sequence, and the target word is explicitly marked inside the context before the text is passed to the model. This preprocessing step helps the model focus on the target item while still using the surrounding learner-specific context (See Table 1).

The dataset is divided into training, development, and test splits for each language group. In our experiments, each L1 contains 6,091 training instances, 677 development instances, and 748 test instances see Table 2. The training and development sets include the gold difficulty scores, while the test set is used for blind evaluation.

System performance is measured mainly with root mean squared error (RMSE). This is the main metric used to compare systems and determine their ranking. RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2},$$

where y_i is the gold difficulty score, $\hat{y}_i$ is the predicted score, and N is the number of test items. A lower RMSE means better performance because it shows that the predicted scores are closer to the reference scores.

This dataset is useful for the task because it allows the model to learn from both the target word and the learner-specific context. It also enables comparison of how vocabulary difficulty varies across Spanish, German, and Mandarin Chinese learners, rather than treating all learners as a single group.

Field	Description	Example
item_id	Unique identifier for each item	9
L1	Learner first language	es
en_target_word	English target word	capacity
en_target_pos	Part of speech of the target word	noun
en_target_clue	Partial clue for the target word	c_____
L1_source_word	Source cue / translation	cabida
L1_context	Learner L1 context	short Spanish prompt
GLMM_score	Gold difficulty score in labelled data	-1.727

Table 1: Main fields used in the shared task dataset, illustrated with a Spanish labelled example.

3 Literature Review

Recent work on difficulty prediction has moved in two closely related directions: item difficulty modeling from assessment content and text difficulty modeling for language learners.

Kapoor et al. (2026) study item difficulty prediction for reading comprehension items using a repository built from standardized tests. Their work shows that difficulty depends on more than the question text alone. They annotate items with linguistic, test, and context features, and then compare these with LLM-based embeddings. Their results show that the best performance comes from combining these sources of information, while text analysis features and LLM embeddings perform similarly when used on their own. This suggests that interpretable linguistic features still remain useful even when modern embedding methods are available.

A related line of work focuses more directly on text difficulty for second language learners. Zhang and Lu (2025) align English text difficulty with CEFR levels for L2 readers and show that lexical, syntactic, and discoursal features can be used to classify texts effectively. Their findings are especially relevant because they show that text difficulty is not explained by traditional readability formulas alone. Instead, features such as age of acquisition, lexical decision time, mean length of T-unit, academic vocabulary, and complex nominals play an important role. They also show that the features that separate lower proficiency levels are not ex-

item_id	L1	en_target_word	L1_source_word	L1_context
6782	Spanish	luck	suerte	Te deseo suerte con tu nuevo trabajo
6771	German	herd	Herde	Die Viehherde graste auf der Weide
6772	Chinese	harm	伤害, 损害	这类项目会损害环境。

Table 2: Illustrative examples showing the dataset format for Spanish, German, and Mandarin Chinese learners.

actly the same as those that matter at higher levels.

Li et al. (2025) examine item difficulty modeling with fine-tuned small and large language models. Their results show that fine-tuned small models such as BERT and RoBERTa remain very competitive, and that careful data augmentation can improve prediction further. In contrast, larger autoregressive models and prompting-based approaches are less effective in their setting, mainly because the task is constrained by small training data and imbalanced difficulty distributions. Their work is important because it shows that stronger results do not always come from using larger models, but from matching the modeling strategy to the structure of the task.

This shared task also builds on a longer line of work on lexical complexity and simplification. Earlier shared tasks, such as SemEval 2016 on Complex Word Identification and the BEA 2018 multilingual CWI task, treated the problem mainly as deciding whether a word is difficult for a reader (Paetzold and Specia, 2016; Yimam et al., 2018). Later tasks moved closer to graded prediction, especially SemEval 2021 Task 1 on Lexical Complexity Prediction, while the BEA 2024 MLSP shared task connected lexical complexity with multilingual simplification pipelines (Shardlow et al., 2021, 2024). Compared with these earlier settings, the present task focuses more directly on learner-specific vocabulary difficulty by taking the learner’s first language and contextual information into account.

Taken together, these studies suggest that difficulty prediction benefits from three broad ideas: the use of learner- or task-relevant linguistic features, the inclusion of contextual information when available, and the careful choice of modeling architecture rather than relying only on model size. Our work follows this direction by combining a strong contextual encoder with a separate regression component, aiming to capture both contextual meaning and structured variation in the prediction of vocabulary difficulty. In particular, the transformer model

provides the main contextual representation of the target word, while the regression branch offers an additional signal derived from dense embedding features. This design keeps the system simple, but still allows it to combine information from complementary sources.

4 Methodology

Our system combines a fine-tuned transformer model with a second-stage regression model built on dense embedding features. The main predictive signal comes from DeBERTa v3 Large (He et al., 2021), which is fine-tuned separately for each learner group. In parallel, we extract feature representations from DeBERTa and XLM-RoBERTa (Conneau et al., 2020) and use them as input to a CatBoost regressor (Prokhorenkova et al., 2018). The final prediction is obtained by combining the transformer score and the CatBoost score through weighted averaging. Figure 1 shows a high-level view of the workflow.

4.1 Preprocessing and Feature Extraction

The input to the transformer is created by combining several text fields defined in the model configuration file. During preprocessing, these fields are merged into a single sequence in a fixed order, separated by the tokenizer’s separator token when available. The target word is explicitly marked inside the context by surrounding it with special tags, `<t>` and `</t>`. This design means that the model does not process the target word in isolation. The remaining unused columns are removed during preprocessing.

For both models, we used an identical set of input features: the L1 source word, L1 context, English part of speech, English clue, and English target word. To support the second-stage regression model, we extract dense vector representations from two pretrained transformers: DeBERTa v3 Large and XLM-RoBERTa Large. The embedding script loads a tokenizer and base transformer model for each encoder, applies tokenization with padding

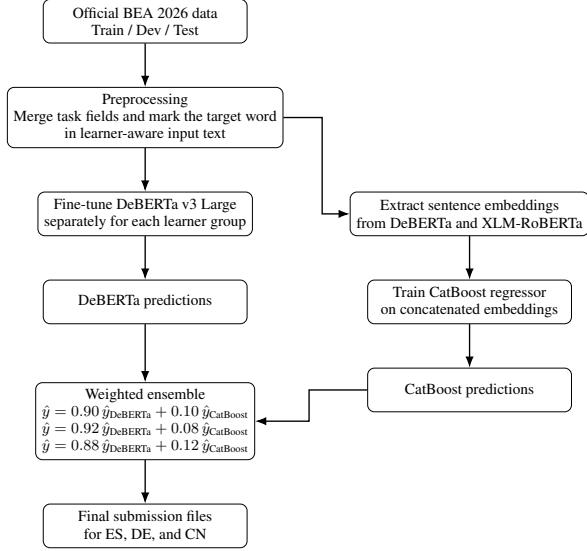


Figure 1: Workflow of the proposed system. DeBERTa v3 Large serves as the main prediction model, while a CatBoost regressor is trained on sentence embeddings extracted from DeBERTa and XLM-RoBERTa. The final prediction is obtained through weighted ensembling. The figure reports the main ensemble setting, 0.90/0.10, together with the two additional weight configurations, 0.92/0.08 and 0.88/0.12, explored in our submitted runs.

and truncation, and then computes a sentence-level representation using mean pooling over the last hidden states.

4.2 Experimental Design

The transformer component is trained using the Hugging Face Trainer API.¹ Tokenization is performed after preprocessing, with truncation applied to the combined input text. Training is carried out with epoch-based evaluation, epoch-based checkpointing, and selection of the best model at the end of training.

DeBERTa, short for Decoding-enhanced BERT with Disentangled Attention, is a transformer-based encoder that models content and positional information separately. This design helps the model capture fine contextual differences more precisely, which is especially useful in vocabulary difficulty prediction where small lexical and contextual cues matter. We trained the DeBERTa model for each language using 6 epochs with a batch size of 16, a learning rate of $2e-5$, and a warmup ratio of 0.1. The weight decay was set to 0.1 for Spanish and Chinese, and 0.0 for German. We kept these settings because they provided a stable training pro-

cess and gave the strongest development performance in our experiments, while still keeping the setup simple and consistent across the three learner groups.

In our final submission, DeBERTa v3 Large serves as the main neural model. Its role is to learn a direct mapping from the learner-aware input representation to the target difficulty score. Since the model is trained separately for each learner group, it can adapt to differences across Spanish, German, and Mandarin Chinese learners without mixing the three data distributions during inference. CatBoost is a gradient boosting regressor designed to model continuous values from structured feature representations. In our system, it serves as a second-stage model over embedding-based features and provides an auxiliary prediction that complements the main transformer output. Additionally, the CatBoost stage uses concatenated DeBERTa and XLM-RoBERTa embeddings as its input features. For each learner group, the code loads the embedding arrays, concatenates the two representation spaces, and fits a CatBoostRegressor using the training labels from the official training split. The model is trained with RMSE as the loss function, 2000 boosting iterations, a learning rate of 0.03, and a tree depth of 6. The CatBoostRegressor operates on fixed embedding vectors and contributes an auxiliary score that complements the main DeBERTa output. This is useful because the final regression problem may involve nonlinear interactions in the representation space that are not fully exploited by a standard transformer regression head.

The final prediction is obtained by combining the DeBERTa score and the CatBoost score through weighted averaging. Before combining the two files, the code checks that the `item_id` ordering matches exactly. The final prediction is then computed as:

$$\hat{y}_{\text{final}} = \alpha \hat{y}_{\text{DeBERTa}} + (1 - \alpha) \hat{y}_{\text{CatBoost}}, \quad (1)$$

where α is the weight assigned to the DeBERTa prediction. For the main run, we used:

$$\hat{y}_{\text{final}} = 0.90 \hat{y}_{\text{DeBERTa}} + 0.10 \hat{y}_{\text{CatBoost}}. \quad (2)$$

We also generated two additional runs using nearby weights, namely 0.92/0.08 and 0.88/0.12, to test whether small changes in the ensemble balance would improve stability or final performance (See Table 4). The script uploaded here corresponds to

¹https://huggingface.co/docs/transformers/main_classes/trainer

the run with weights 0.88 and 0.12, but the same logic was used for all three submitted runs.

5 Results

This section presents the performance of the models explored in our experiments and compares them with the final ensemble system. We first report the development results of the main transformer models that were considered during model selection. We then present the official results of the three submitted ensemble runs, which were obtained by combining DeBERTa v3 Large and CatBoost with slightly different weights.

5.1 Transformer Models

Table 3 compares the three main transformer settings that were examined during development: XLM-RoBERTa Base, XLM-RoBERTa Large, and DeBERTa v3 Large. The results show a clear and consistent pattern across all three learner groups. XLM-RoBERTa Large improved substantially over XLM-RoBERTa Base, which suggests that increased model capacity was useful for this task. However, the strongest results were obtained with DeBERTa v3 Large in every language.

The advantage of DeBERTa was especially clear in Chinese, where it reached the lowest RMSE among the three transformer settings. The same trend also appeared in Spanish and German. This consistent improvement made DeBERTa the strongest candidate for the main prediction branch of the final system.

Model	L1	RMSE ↓	Pearson ↑
XLM-RoBERTa Base	Spanish	1.380	0.708
XLM-RoBERTa Base	German	1.383	0.710
XLM-RoBERTa Base	Chinese	1.227	0.711
XLM-RoBERTa Large	Spanish	1.161	0.828
XLM-RoBERTa Large	German	1.153	0.824
XLM-RoBERTa Large	Chinese	1.061	0.820
CatBoost	Spanish	1.737	0.423
CatBoost	German	1.698	0.375
CatBoost	Chinese	1.464	0.499
DeBERTa v3 Large	Spanish	1.021	0.854
DeBERTa v3 Large	German	1.059	0.838
DeBERTa v3 Large	Chinese	0.925	0.841

Table 3: Development set results of the main transformer models across the three learner groups. Lower RMSE and higher Pearson indicate better performance.

5.2 Final Ensemble

We submitted three runs with nearby weight settings. The main run used a weight of 0.90 for DeBERTa and 0.10 for CatBoost. Two additional runs, 0.92/0.08 and 0.88/0.12, were included to test whether a small change in the ensemble balance would improve the final performance. The results are shown in Table 4.

Run	Weights (D/C)	Spanish	German	Chinese
Run 1	0.90 / 0.10	1.040	0.992	0.883
Run 2	0.92 / 0.08	1.040	0.993	0.882
Run 3	0.88 / 0.12	1.041	0.992	0.885
Best Model	- / -	0.903	0.885	0.776

Table 4: Official RMSE results of the three submitted ensemble runs, compared with the best model submitted in the shared task. D/C denotes the weights assigned to DeBERTa and CatBoost, respectively. Lower values are better.

The final ensemble results were very close across the three runs. This is consistent with the development results, where DeBERTa was already the strongest standalone model.

6 Discussion

The gap between the standalone DeBERTa model and the final ensemble is much smaller. This is also an important finding. It suggests that most of the predictive signal already comes from DeBERTa, while CatBoost provides a limited but useful correction. The small differences across the three submitted ensemble runs support the same interpretation. If the gains had been random, the relative order of the runs would likely have varied more sharply across languages. Instead, the scores remain very close, which points to a stable system with a fairly fixed error profile (See Table 4). Overall, the results show that the final system benefited more from a strong contextual encoder than from aggressive reweighting in the ensemble stage. The submitted runs confirmed that the proposed hybrid approach was stable across languages, with the strongest final performance observed for Chinese and the most difficult condition appearing in Spanish.

A likely explanation for the gap between the weaker models and the final model is the way they represent lexical information. Vocabulary difficulty prediction is not only a matter of general sentence meaning. The model must detect subtle differences

in target-word usage, clue structure, and learner-specific context, then map that information to a continuous score. Smaller or less task-suited encoders may capture broad semantic information, but they are more likely to smooth over the fine distinctions that matter in this setting. DeBERTa appears to reduce this problem, and the CatBoost component adds a second decoding layer over dense embeddings, which helps recover some structured patterns that may be missed by a simple transformer regression head.

From a hypothesis-testing perspective, several item-level factors are worth examining in future work. The first is *word length*, since longer words may be more morphologically complex and therefore harder both for learners and for the model. The second is *frequency*, because rare words are usually more difficult and may also be less well represented in pretrained models. The third is *part of speech*, as verbs, adjectives, and abstract nouns may not behave in the same way. A fourth factor is *context dependence*: some words can be understood from the clue alone, while others require close reading of the full prompt. These factors are promising because they offer a concrete way to test whether the model’s residual errors are concentrated in specific lexical categories rather than distributed randomly across the dataset.

7 Conclusion

In this paper, we have shown that combining DeBERTa v3 Large with a CatBoost regressor trained on embeddings extracted from DeBERTa v3 Large can provide good estimates of the difficulty of target English words and incorporate additional user-specific data into consideration. Among the transformers examined in the course of our study, DeBERTa v3 Large showed the best performance on its own, outperforming XLM-RoBERTa Base and XLM-RoBERTa Large in Spanish, German, and Chinese. The presented ensemble confirmed the stability of a simple linear combination of an effective contextual language model and a relatively lightweight machine learning model. We showed that the effectiveness of the system is largely dependent on the system’s ability to capture subtle lexical and contextual information about the given input. Second, the improvement comes not merely from increasing the size of the system or adding layers, but from the careful integration of complementary signals in the process. In our case, the

main power was provided by DeBERTa, whereas CatBoost added a minor correction that improved predictions. The small gap between the submitted runs shows that the system was relatively stable in spite of changing ensemble weights.

However, there are also several important limitations. While Chinese was the most successfully predicted language, Spanish was the most difficult one, indicating that the link between the learner, contextual information, and word difficulty may be differently tractable across learner populations. An important direction of future research would consist of analyzing errors at the level of individual items in order to find out whether additional explicit lexical features help in the predictions.

Acknowledgments

This research was funded by the Challenge-Based Research Funding Program, Grant No. I030-IFE002-C2-T1-E, of the Institute for the Future of Education, Tecnológico de Monterrey, Mexico. The work was also carried out with partial support from Grant No. 20260626 (G.S.) awarded by the Secretaría de Investigación y Posgrado (SIP) of Instituto Politécnico Nacional, Mexico.

References

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the bea 2026 shared task on vocabulary difficulty prediction for english learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [Deberta: Decoding-enhanced BERT with disentangled attention](#). In *International Conference on Learning Representations*.
- Radhika Kapoor, Sang T. Truong, Nick Haber, Maria Araceli Ruiz-Primo, and Benjamin W. Domingue. 2026. Prediction of item difficulty for reading comprehension items by creation of annotated item repository. *arXiv preprint arXiv:2502.20663*.

- Ming Li, Hong Jiao, Tianyi Zhou, Nan Zhang, Sydney Peters, and Robert W Lissitz. 2025. Item difficulty modeling using fine-tuned small and large language models. *Educational and Psychological Measurement*, 85(6):1065–1090.
- 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.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. [Catboost: unbiased boosting with categorical features](#). In *Advances in Neural Information Processing Systems*.
- 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 Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*. 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.
- Xiaopeng Zhang and Xiaofei Lu. 2025. [Aligning linguistic complexity with the difficulty of english texts for l2 learners based on cefr levels](#). *Studies in Second Language Acquisition*, 47(5):1407–1434.

RETUYT-INCO at BEA 2026 Shared Task 1: Feature-Enriched mDeBERTa for Word Difficulty Prediction

Santiago Robaina and Luis Chiruzzo and Aiala Rosá

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

Abstract

We describe the RETUYT-INCO participation in the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners, a regression task that predicts GLMM psychometric difficulty scores for English target words given an L1 cue (Spanish, German, and Mandarin). We submitted two systems to the closed track (which restricts participants to the provided shared-task data and standard NLP resources, excluding external corpora and large language models): a feature-engineered XGBoost regressor for all three L1s, and, for Spanish, a 3-seed ensemble of mdeberta-v3-base fine-tuned with the same handcrafted features prepended as input text tokens. Our best test result is 1.094 RMSE on Spanish (ensemble), a 13.0% reduction over the XLM-RoBERTa-base closed baseline. We highlight two findings. First, a LaBSE cross-lingual cosine between the L1 source word and the English target word is the largest single-feature addition in our incremental ablation, reducing average development-split (dev) RMSE by 0.091 on top of an already strong string/frequency/POS feature set. Second, feature-only XGBoost, with no neural fine-tuning and no GPU, already beats the XLM-RoBERTa-base closed-track development baseline on average across the three L1s (1.273 vs. 1.287 RMSE).

1 Introduction

Predicting how hard an English word is for a second-language learner is a core problem for intelligent tutoring, adaptive reading, and vocabulary tools. The BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners (Felice and Skidmore, 2026) casts this as a regression problem: given an English target word and an L1 cue (source-language word, context sentence, and a partial orthographic clue), predict a continuous Generalized Linear Mixed Model (GLMM) psychometric difficulty score. The task covers three L1s (Spanish,

German, and Mandarin) and two tracks (closed and open). The closed-track baseline is a fine-tuned XLM-RoBERTa-base (Conneau et al., 2019).

The task sits at the intersection of two traditions. On the NLP side, the Complex Word Identification shared tasks (Paetzold and Specia, 2016; Yimam et al., 2018) framed word-level difficulty as classification by non-native readers; SemEval 2021 (Shardlow et al., 2021) replaced this with a continuous Lexical Complexity Prediction target, and BEA 2024 MLSP (Shardlow et al., 2024) extended the setting to multilingual lexical simplification. Handcrafted features very similar to the ones we use here (word length, frequency, morphology, character n -grams) have been standard baselines across this line of work, with fine-tuned neural encoders more recently taking the lead. On the psychometric side, vocabulary testing research long predates these benchmarks (Laufer and Goldstein, 2004), and the Knowledge-based Vocabulary Lists project (Schmitt et al., 2021, 2024) provides the response data that underlies the BEA 2026 targets; the GLMM difficulty scores follow the random-item IRT tradition (De Boeck, 2008; Dunn, 2024). The most direct predecessor is Skidmore et al. (2025), who benchmark transformer architectures for predicting vocabulary-test item difficulty on a KVL-derived corpus; their XLM-RoBERTa setup is adapted as the BEA 2026 closed-track baseline.

We approach the task under a self-imposed lightweight constraint: we target systems that can be trained and served on a single low-end GPU or on CPU, and we restrict ourselves to the closed track. This reflects the compute realities of the Global South, where access to large clusters and API-gated models is limited, and the privacy constraints of on-device educational deployments. We submit two systems for the closed track: (i) an XGBoost (Chen and Guestrin, 2016) regressor over a set of handcrafted string, morphological, POS, fre-

quency, and cross-lingual semantic-similarity features, for all three L1s, and (ii) for Spanish only, a 3-seed ensemble of mdeberta-v3-base (He et al., 2023) fine-tuned for regression with the computed features prepended as input text tokens.

Our research question centres on which handcrafted features carry the most signal for cross-lingual vocabulary difficulty prediction. Two findings emerge. First, a LaBSE (Feng et al., 2022) cross-lingual cosine between the L1 source word and the English target word is the largest single-feature drop we observe in an incremental-addition ablation (Table 1), reducing average development-split RMSE by 0.091 across ES/DE/CN in our tree-based pipeline (§3) and adding a further 0.017 ES development-split RMSE when also prepended as an input token to mDeBERTa (§4). We note that incremental ablations attribute shared variance to whichever feature is introduced first, so this should be read as a lower bound on its marginal contribution rather than a proof of strict dominance over every other single feature; see §3. Second, feature-only XGBoost over our full feature set beats the XLM-RoBERTa-base closed-track development baseline on average (1.273 vs. 1.287), with no neural fine-tuning and no GPU. The official-test picture is mixed: XGBoost beats the test baseline on Mandarin, is within noise on German, and underperforms on Spanish (§5). The mDeBERTa ensemble is reported as our submitted best system, not as an efficiency contribution: it is the same total size as the XLM-RoBERTa baseline per seed, and approximately triples inference cost as an ensemble.

2 Dataset and Task

The task provides, per L1, 6,091 training items and 677 development items, with a held-out test set released at submission time (Felice and Skidmore, 2026). Each item has an English target word and its part of speech, a partial orthographic clue of the target, an L1 translation, and an L1 context sentence. The target is a continuous GLMM psychometric difficulty score, ranging roughly from -6 to +5 with lower values indicating harder items. The primary metric is RMSE, with Pearson correlation as a secondary metric. Two tracks are defined: closed (only the provided data and standard NLP resources) and open (external data and LLMs allowed). We participate in the closed track only; the open track was out of scope under our lightweight

constraint.

3 Feature Engineering and XGBoost

3.1 Feature set

We extract a set of handcrafted features per item, grouped in this section by type rather than by order of addition. All features are computed on CPU.

Target-word features (English only): word length, one-hot POS over the eight task categories, Zipf word frequency from the wordfreq package (Speer, 2022), indicators for common English suffixes (-tion, -sion, -ing, -ence, -ment, -able, -ness, -ly) and prefixes (dis-, re-, un-, sub-, out-, over-), maximum consonant cluster length, double-letter indicator, syllable count, and an English latinate-root indicator.

Source-word features (L1 only): source-word length, source-word syllable count, multiword-source indicator, and an accent indicator that matches Spanish and German diacritics (zero for CN).

Cross-lingual string similarity (both words): length difference and ratio, normalized Levenshtein distance, character-bigram Jaccard overlap, Jaro-Winkler similarity (Winkler, 1990), shared prefix and suffix ratio, a first-letter match indicator, and a suffix-transform match. The suffix-transform rules are hand-written per L1 and encode the expected cognate shape (e.g. *-tion* → *-ción* in ES, 15 rules total; *-tion* → *-ierung*, *-ty* → *-tät*, ... in DE, 18 rules total; empty for CN).

Cross-lingual semantic similarity: embed_cosine, the cosine similarity between LaBSE (Feng et al., 2022) sentence embeddings of the English target word and the L1 source word. This is the single largest RMSE drop in our incremental feature ablation (§3) and is the main narrative thread of this paper.

Context: length of the L1 context sentence in characters.

3.2 Model and protocol

We fit an XGBoost regressor on the training split and report RMSE on the development split. Rows 1–3 of Table 1 use linear regression as a deliberately simple baseline for small, near-monotone feature sets; from row 4 onward the features introduce non-linear effects and interactions that motivate XGBoost (`n_estimators=300`, `max_depth=5`, `learning_rate=0.05`), which is also the model used in our submission. The final

row (row 7) uses L1-tuned XGBoost parameters from an 80-iteration random search with 5-fold CV on the training split; tuning moved dev RMSE by only ~ 0.001 , so feature engineering dominated.

3.3 Observations

Table 1 walks up our feature set. The largest single-feature *incremental* addition is `embed_cosine`, which drops average dev RMSE by 0.091 (row 5 $\rightarrow$ row 6). The effect is largest on Mandarin (CN dev RMSE 1.257 $\rightarrow$ 1.181, -0.076), which is consistent with CN having no meaningful orthographic signal between the L1 and English; LaBSE provides the missing cross-lingual bridge. A caveat on attribution: row 5 bundles three features (Jaro–Winkler, char n -gram overlap, Zipf frequency) and produces a larger combined drop (-0.151) than `embed_cosine` alone; because incremental-addition ablations credit shared variance to whichever feature is introduced first, we cannot rule out that one of those three, introduced last, would match or exceed `embed_cosine`’s marginal effect. What we can say is that `embed_cosine` retains a 0.091 average improvement after a feature set that already captures cognate orthography and frequency—i.e., it is not redundant with them. On ES and DE, `embed_cosine` still gives the largest single drop *in this ordering*, alongside target-word Zipf frequency and cross-lingual Jaro–Winkler similarity, which exploit cognate structure.

By row 7, feature-only XGBoost reaches 1.273 average dev RMSE, below the XLM-RoBERTa-base closed dev baseline (1.287), on CPU and with no neural fine-tuning. A leave-one-out ablation on the 9 ES-targeted v2 features confirmed each contributes positively (individual deltas 0.0004–0.0057 on 5-fold CV); we omit per-feature details for space. A full drop-one ablation of `embed_cosine` from the v2 feature set would tighten this claim and is left for future work (§7).

¹Tuned parameters for row 7: ES uses `n_est=600`, `depth=3`, `lr=0.02`, `subsample=0.85`, `colsample=0.70`, `alpha=0.5`, `lambda=2.0`; DE and CN use `n_est=400`, `depth=3`, `lr=0.05`, `subsample=0.85`, `colsample=0.85`, `alpha=0.1`, `lambda=2.0`. The row 7 / row 6 delta therefore conflates the v2 feature block with the L1-tuned hyperparameters. The v2 feature block itself is applied to all three L1s, with per-L1 rule sets (15 suffix-transform rules for ES, 18 for DE, empty for CN so the suffix-transform component is inert there; the remaining v2 features apply equally to all three L1s).

4 mDeBERTa with Feature-Enriched Input

For Spanish, we fine-tune `microsoft/mdeberta-v3-base` (He et al., 2023) for regression (`num_labels=1`). The full model is approximately 276M parameters (an 86M transformer backbone plus a ~ 190 M 250K-token word-piece embedding matrix), essentially the same total size as the XLM-RoBERTa-base closed baseline (~ 270 M).

Input format. We prepend five feature tokens to the encoder input, then concatenate the raw L1 and English fields with [SEP] markers:

```
wlen=N | nedit=N | pos=X | clue=N |
esim=N | L1_word [SEP] L1_context [SEP]
en_clue [SEP] en_word
```

The five prepended features (English word length, normalised edit distance, POS, clue length, and `esim`) are a subset of the XGBoost feature set rendered as plain text. `esim` is the LaBSE cosine of §3 (`embed_cosine`), now available to the encoder as a numeric input token.

Training. GLMM scores are normalised to zero mean and unit variance at training time and denormalised before RMSE computation. We train with learning rate 2×10^{-5} , a cosine schedule with 10% warmup, up to 10 epochs with early stopping (patience 3), batch size 32, weight decay 0.01, fp16, and max sequence length 256. Training was performed on a single Google Colab T4 GPU.

Ensemble. We train three seeds ($\{10, 42, 123\}$) and average their predictions. The ensemble triples inference compute and memory: three ~ 276 M-parameter models are loaded in parallel, for approximately 828M parameters and $3 \times$ forward passes per prediction. Each individual seed already beats the XLM-RoBERTa-base baseline on ES dev (Table 2) by 0.179–0.220 RMSE; the ensemble is a further refinement of 0.034 RMSE over the best individual seed (seed 42, 1.137) and 0.059 RMSE over the mean of the three seeds, so it is a refinement on top of the per-seed gain rather than the source of the gap to the baseline.

Why Spanish only. We did not extend the mDeBERTa ensemble to all three L1. DE and CN ensemble were skipped for time reasons and our own limitations of knowledge of the languages. Our DE and CN submission is therefore XGBoost only.

Test-time retraining. The submitted ES ensemble is retrained on train+dev combined before predicting the test set. The test RMSE (1.094) is

Feature set	Model	ES	DE	CN	Avg	Δ vs. prev
1. word_len	LR	1.778	1.707	1.509	1.665	—
2. + edit distance	LR	1.705	1.596	1.487	1.596	−0.069
3. + POS	LR	1.663	1.536	1.493	1.564	−0.032
4. + morphology + shared prefix + context	XGBoost	1.613	1.524	1.459	1.532	−0.032
5. + Jaro–Winkler + n -gram + Zipf freq.	XGBoost	1.464	1.421	1.257	1.381	−0.151
6. + embed_cosine (LaBSE)	XGBoost	1.350	1.339	1.181	1.290	−0.091
7. + L1-specific (v2)	XGBoost ¹	1.327	1.334	1.158	1.273	−0.017
XLM-RoBERTa-base (closed) baseline	XLM-R (~270M)	1.357	1.328	1.175	1.287	—

Table 1: Feature progression on dev (RMSE, 3 dp). Rows 1–3 add the target-word length, a first cross-lingual string feature (edit distance), and target POS. Row 4 adds target-word morphology (suffix/prefix indicators, consonant cluster, double-letter) together with the shared-prefix ratio and context length. Row 5 adds further cross-lingual string similarity (Jaro–Winkler, character-bigram overlap, first-letter match) and target-word Zipf frequency. Row 6 (bold) adds the cross-lingual semantic similarity (embed_cosine). Row 7 adds the L1-specific v2 block and switches to L1-tuned XGBoost hyperparameters. The last row of the table is the task baseline for reference.

therefore not directly comparable to the dev RMSE of the same configuration (1.103).

5 Results and Analysis

On Dev-set. Feature-only XGBoost beats the XLM-RoBERTa-base closed dev baseline on average and on ES and CN individually; on DE it is within noise of the baseline (1.334 vs. 1.328, +0.006). See Table 2. The pipeline uses CPU-fit trees and no neural fine-tuning. The submitted ES ensemble reaches 1.103 dev RMSE, a further 0.224 RMSE below our XGBoost row. This gap is not an efficiency result, since per-seed mDeBERTa has essentially the same parameter count as XLM-RoBERTa-base; it is an effect of the feature-enriched input format, target scaling, and 3-seed averaging. Each individual seed already beats the XLM-RoBERTa dev baseline on ES.

On Test-set. On the official closed-track test leaderboard (Table 3), our ES ensemble reports 1.094 RMSE / 0.843 Pearson, placing us 9th of 21 participating teams on Spanish (best-per-team collapse), and reducing the test-baseline RMSE by 13.0%. XGBoost on Mandarin also beats the test baseline (1.106 vs. 1.140, −3.0%), placing us 18th of 21 teams. German XGBoost is within noise of the test baseline (1.260 vs. 1.258, +0.2%, rank 18/20). ES XGBoost underperforms the test baseline (1.323 vs. 1.257, +5.3%), despite beating the dev baseline (1.327 vs. 1.357). This dev/test inversion on ES is the clearest sign of distribution

shift between the two splits for our feature-based system.

6 Conclusions

Among the handcrafted features we tested, the LaBSE cross-lingual cosine between the L1 source word and the English target word carries the most task-relevant signal: adding it reduces average dev RMSE by 0.091 across ES/DE/CN in our XGBoost pipeline, and adds a further 0.017 ES dev RMSE when also included as a prepended input token to mDeBERTa. Feature-only XGBoost, with no GPU, already beats the XLM-RoBERTa-base closed dev baseline on average (1.273 vs. 1.287). Our submitted ES system, a 3-seed mDeBERTa ensemble, places 9/21 on the closed-track ES test leaderboard at 1.094 RMSE.

7 Limitations

The neural ensemble was run for ES only; DE was skipped for compute reasons and CN was skipped by design (Latin-script feature assumptions). The submitted ES ensemble loads ~828M parameters and runs three forward passes per prediction: it is not an efficient system relative to the baseline, and we report it as our best test result rather than as an efficiency contribution. We prioritise breadth over depth and did not run per-L1 mDeBERTa hyperparameter searches. The ES XGBoost shows a clear dev/test inversion that we do not fully explain. We did not explore open-track resources. Finally, our “largest single-feature incremental drop” claim for embed_cosine rests on incremental-addition evidence and is therefore order-dependent: the three features added in row 5 of Table 1 (Jaro–Winkler, char n -gram, Zipf) share credit for a combined

²The submitted XGBoost boosters were not persisted; their serialised size on disk was not measured before submission. The model is a gradient-boosted tree ensemble (up to 600 trees of depth 3) trained on ≤ 23 numeric features.

System	Params (total)	ES	DE	CN
XLM-RoBERTa-base (closed baseline)	~270M	1.357	1.328	1.175
full_xgb_v2_embed (our XGBoost, submitted)	tree ensemble (CPU) ²	1.327	1.334	1.158
mDeBERTa (seed 10)	~276M	1.178	—	—
mDeBERTa (seed 42)	~276M	1.137	—	—
mDeBERTa (seed 123)	~276M	1.172	—	—
mDeBERTa 3-seed ensemble (submitted, ES only)	~828M at inference	1.103	—	—

Table 2: Development-split RMSE (3 dp). Per-seed Pearson on ES is 0.789 / 0.821 / 0.826; the ensemble reaches 0.827. The mDeBERTa backbone (86M) and the XLM-RoBERTa backbone (~85M) are essentially the same size; total parameters including the $250K \times 768$ word-piece embedding matrix are ~276M and ~270M respectively.

L1	System	RMSE	Pearson	Team rank	Δ RMSE vs. baseline
ES	mDeBERTa 3-seed ensemble (submitted)	1.094	0.843	9 / 21	-0.163 (-13.0%)
ES	XGBoost (secondary submission)	1.323	0.713	—	+0.066 (+5.3%)
DE	XGBoost (submitted)	1.260	0.713	18 / 20	+0.002 (+0.2%)
CN	XGBoost (submitted)	1.106	0.754	18 / 21	-0.034 (-3.0%)
ES	XLM-RoBERTa-base (closed test baseline)	1.257	0.765	—	—
DE	XLM-RoBERTa-base (closed test baseline)	1.258	0.773	—	—
CN	XLM-RoBERTa-base (closed test baseline)	1.140	0.753	—	—

Table 3: Official test-set results, closed track. A negative Δ RMSE indicates our system is lower than (better than) the baseline. Team rank collapses each team to its best submission per L1; dashes in that column indicate a row shown for comparison only and not counted separately (our ES team rank is determined by the ensemble). The ES XGBoost row is reported to contextualise the dev/test behaviour of the feature-only system.

—0.151 drop, and a drop-one ablation from the full v2 set would be needed to establish strict per-feature dominance. We leave that ablation to future work.

Acknowledgments

This paper was funded by *Agencia Nacional de Investigación e Innovación* (ANII, Uruguay), Projects *FSSED_2_2023_1_179355* and *FMV_1_2023_1_176581*.

References

- 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.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#). *CoRR*, abs/1911.02116.
- Paul De Boeck. 2008. Random item IRT models. *Psychometrika*, 73(4):533–559.
- Karen J. Dunn. 2024. Random-item Rasch models and explanatory extensions: A worked example using L2 vocabulary test item responses. *Research Methods in Applied Linguistics*, 3(3):100143.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications*. To appear.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Ariavazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891. 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. *arXiv preprint arXiv:2111.09543*.
- Batia Laufer and Zahava Goldstein. 2004. Testing vocabulary knowledge: Size, strength, and computer adaptiveness. *Language Learning*, 54(3):399–436.
- 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.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. In-

- roducing knowledge-based vocabulary lists (KVL). *TESOL Journal*, 12(4).
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based Vocabulary Lists*. University of Toronto Press, Toronto.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Sagion. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. Transformer architectures for vocabulary test prediction. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Robyn Speer. 2022. [rspeer/wordfreq: v3.0](#).
- William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*.
- 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.

Token Titans at BEA 2026 Shared Task 1: Multilingual Lexical Complexity Prediction via Fine-Tuned XLM-RoBERTa with Ensemble Decoding

Anubhab Parashar and Sandeep Mathias

Presidency School of Computer Science and Engineering

Presidency University, Bangalore

{anubhab.20231COM0043, sandeepalbert}@presidencyuniversity.in

Abstract

In this paper, we describe our submission to the BEA 2026 Shared Task on Multilingual Lexical Complexity Prediction. Our system fine-tunes XLM-RoBERTa Large separately for Spanish, German, and Chinese, feeding each instance as a concatenation of the source word, its context in a sentence, an English clue, and the English target word. Training uses Z-score label normalization and two independent runs that differ in learning rate, random seed, etc. On the official test set, the system scores RMSE = 1.170 and Pearson Correlation = 0.812.

1 Introduction

Estimating how hard a word is to read depends on who is doing the reading. A word which is easy for a native speaker to understand may be quite complex for a language learner, and the same word can sit at opposite ends of the difficulty scale depending on its context. Lexical complexity prediction (LCP) formalizes this as a regression task: given a word in sentence context, predict a score reflecting how difficult it is for a reader.

Most LCP work has concentrated on English (Shardlow et al., 2021; North et al., 2022). The BEA 2026 Shared Task extends this to learners whose first language (L1) is Spanish, German, or Chinese – three languages that differ substantially in morphology, script, and how well-represented they are in multilingual pretrained models. Chinese is the most challenging of the above languages—no whitespace segmentation, character-level ambiguity, and thinner coverage in most encoders than either of the Latin-script languages.

The BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners requires participants to predict the complexity of a word for learners of English from different L1 backgrounds – namely Spanish, German and Chinese. Each instance in the task also includes an English clue

word and an English target alongside the L1 word and context. A model with strong English representations can potentially use the clue and target as anchors even when the source language is less familiar. The dataset in the shared task provides us with the source word, context, clue and target, which can help the classifier in predicting the complexity.

We fine-tune XLM-RoBERTa Large (Conneau et al., 2020) separately for each language, concatenating all four fields into a single flat input string. To squeeze a bit more out of the model without additional data, we train two runs per language under slightly different hyperparameters and improve model performance via weighted averaging of predictions. The final system scores RMSE as 1.170 and Pearson $r = 0.812$ on the test set.

2 Related Work

Lexical complexity prediction. Early readability work used word frequency and syllable counts as proxies for lexical difficulty (Dale and Chall, 1948). The SemEval-2021 LCP shared task (Shardlow et al., 2021) shifted the benchmark toward fine-grained regression on words in context, and transformer-based systems pulled well ahead of feature-engineered pipelines – less because of architectural creativity than because model scale simply helps (North et al., 2022; Zaharia et al., 2022).

Multilingual LCP has received less attention. Most multilingual efforts swap in a multilingual encoder and, sometimes, language-specific lexical resources such as frequency lists or morphological analyzers (Shardlow et al., 2022). An earlier shared task on multilingual lexical simplification was organized during BEA 2024 (Shardlow et al., 2024).

Multilingual pretrained encoders. XLM-RoBERTa (Conneau et al., 2020) covers 100 languages and has become the default backbone

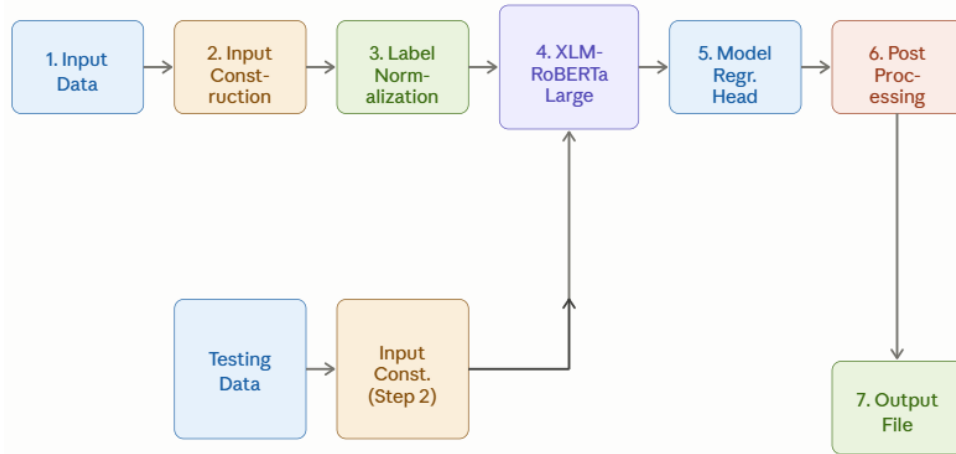


Figure 1: Overview of the proposed pipeline for multilingual lexical complexity prediction.

for cross-lingual classification and regression. In this Shared Task, the standard setup - linear head on [CLS], full fine-tuning - works well enough that there is little reason to deviate from it.

Ensembling. Averaging predictions across runs with different seeds or hyperparameters reduces variance without requiring additional data or a larger model (Lakshminarayanan et al., 2017; Liu et al., 2019). We train two runs per language and combine them with a fixed weighted average (0.6/0.4), with weights chosen on the validation set.

3 System Description

Figure 1 shows our system architecture. For training, we do input construction and label normalization which we feed as input to XLM-RoBERTa to learn our model. At testing time, we also construct the input representations as input to XLM-RoBERTa and the model. The output from the model is sent to a regression head, followed by post-processing to generate our output labels.

3.1 Input Construction

Each instance has four fields: the L1 source word, its sentence context, an English clue word, and an English target word. We join them with `</s>` separators, which XLM-RoBERTa treats as segment boundaries. All four fields are concatenated as:

```
L1_word </s> L1_context </s> en_clue </s>
                               en_target
```

For Chinese, consecutive whitespace is collapsed with a regex pass before tokenization; without it, the tokenizer splits character-segmented text erratically.

3.2 Model

We fine-tune XLM-RoBERTa Large (Conneau et al., 2020) with a linear regression head on the [CLS] token. The full network is updated end-to-end. We train separate models for each language rather than a single joint model; the three languages differ enough in script and morphology that joint training hurt validation RMSE in preliminary experiments.

GLMM scores are z-score normalized before training:

$$\hat{y} = \frac{y - \mu}{\sigma} \quad (1)$$

where μ and σ are computed over the training split for each language. Predictions are denormalized before submission and clipped to $[0.05, 5.0]$; the lower clip handles a small number of predictions that collapsed near zero during early stopping.

3.3 Training

All models use AdamW with fp16 mixed precision, gradient checkpointing, 500 warmup steps, and gradient norm clipped at 1.0. Batch size is 4 with 2 gradient accumulation steps (effective batch size 8), trained for 5 epochs. The checkpoint with the lowest validation RMSE is kept.

3.4 Ensembling

We train two runs per language with different hyperparameters (Table 1). Spanish Run 1 produced the strongest single result in the submission, and its settings were the starting point for German and Chinese - we adjusted learning rate and seed between runs to test whether that configuration would hold. It did not, which is discussed in Section 4.

For German only, we combined Run 1 and Run 2 via weighted averaging:

$$\hat{y}_{\text{ens}} = 0.6 \cdot \hat{y}_1 + 0.4 \cdot \hat{y}_2 \quad (2)$$

The weights were set on the validation set across all languages and not tuned further on a per-language basis.

Lang.	Setting	Run 1	Run 2
es / de	Learning rate	1e-5	1.1e-5
	Scheduler	linear	cosine
	Seed	42	1337
	Max length	384	320
cn	Learning rate	1.2e-5	1.1e-5
	Scheduler	cosine	cosine
	Seed	42	1337
	Max length	320	320

Table 1: Per-language hyperparameters. The following hyperparameters were shared across all runs: AdamW, fp16, batch size 4, 2 gradient accumulation steps, 500 warmup steps, 5 epochs, max_grad_norm 1.0.

3.5 Experimental Setup

We evaluate on the official BEA 2026 test set using RMSE (primary) and Pearson r (secondary). Training and validation splits follow the shared task data release. No additional training data or external resources are used. We compare against the shared task baseline, which fine-tunes xlm-roberta-base with batch size 32, learning rate 3×10^{-5} , and dropout 0.1 on the same four input fields; baseline results are reported on the test set.

4 Results and Analysis

Spanish Run 1 achieved the strongest result in the submission (RMSE = 1.170, Pearson Correlation = 0.812). Spanish Run 2 collapsed entirely—Pearson of 0.043 indicates the model’s predictions are near-random, likely due to training instability under the cosine schedule with seed 1337 on the Spanish data distribution. This is consistent with the instability findings discussed in Section 4.1; the same seed and scheduler combination that worked for German produced a degenerate run for Spanish. Run 2 was excluded from the Spanish ensemble for this reason.

For German and Chinese, both runs completed normally. Run 1 is the weakest of the two runs for German on Pearson correlation, and for Chinese the cosine-scheduled Run 2 achieves notably higher Pearson ($r = 0.777$ vs. 0.718) despite slightly

worse RMSE. Ensemble predictions were generated for German only, where combining the two runs matched Run 2 exactly (RMSE 1.569, $r = 0.772$), suggesting the linear-schedule run added no useful signal for that language. The overall best test result is RMSE 1.170 and Pearson $r = 0.812$ as reported in Table 2.

4.1 Effect of Model Scale

xlm-roberta-large has 560M parameters against 270M for the base variant (Conneau et al., 2020). On large fine-tuning sets, the larger model reliably wins. On smaller datasets the picture is less clean—more parameters means more capacity to overfit, and the effect is sharper when the learning rate is low and the batch size is small.

Both conditions apply here. Our effective batch size is 8 (batch 4, gradient accumulation 2) against the baseline’s 32, and our learning rates range from 1×10^{-5} to 1.2×10^{-5} against the baseline’s 3×10^{-5} . A larger batch size averages out more gradient noise per step, which acts as implicit regularisation; a higher learning rate moves the model away from sharp minima faster. The baseline has both. We have neither—and on German and Chinese, where the training splits are smaller than Spanish, the baseline’s implicit regularisation appears to matter more than the capacity advantage of the larger model.

This is consistent with findings in low-resource transfer learning more broadly (Mosbach et al., 2021). Fine-tuning large pretrained models on small datasets is unstable: performance varies significantly across seeds, the best checkpoint is sensitive to when training stops, and the gap between a good run and a bad one is wider than it is for smaller models. Our Chinese results illustrate this directly—Run 1 and Run 2 differ only in learning rate (1.2×10^{-5} vs. 1.1×10^{-5}) and seed, yet their Pearson correlations are 0.718 and 0.777 respectively, a gap of 0.059 on what should be a small hyperparameter change.

One correction that would likely help is adding explicit regularisation to the regression head—dropout before the linear layer, or L2 weight decay on the head parameters specifically. The baseline uses dropout 0.1 throughout. Our runs use no explicit dropout on the head, relying entirely on the pretrained weights and early stopping to prevent overfit. For a 560M parameter model on a few thousand training examples, that is not enough.

Language	System	RMSE ↓	Pearson r ↑
Spanish	Baseline (xlm-roberta-base)	1.257	0.765
	Run 1 (linear, seed 42)	1.170	0.812
	Run 2 (cosine, seed 1337)	1.885	0.043
German	Baseline (xlm-roberta-base)	1.258	0.773
	Run 1 (linear, seed 42)	1.564	0.760
	Run 2 (cosine, seed 1337)	1.569	0.772
	Ensemble (0.6/0.4)	1.569	0.772
Chinese	Baseline (xlm-roberta-base)	1.140	0.753
	Run 1 (cosine, seed 42)	1.382	0.718
	Run 2 (cosine, seed 1337)	1.436	0.777

Table 2: Per-language results compared against the shared task baseline (xlm-roberta-base). All figures are on the test set. Bold marks the best result per language per metric.

4.2 Discussion

Table 2 compares our runs against the shared task baseline, which uses xlm-roberta-base with a batch size of 32 and learning rate of 3×10^{-5} . The results split clearly by language.

For Spanish, both metrics improve over the baseline—RMSE drops from 1.257 to 1.170 and Pearson rises from 0.765 to 0.812. Upgrading from xlm-roberta-base to xlm-roberta-large and halving the learning rate proved effective for Spanish, where the training data is likely well represented in the larger model’s pretraining corpus.

The baseline RMSE of 1.258 for German is substantially better than our best result of 1.564, and for Chinese, the baseline (1.140) outperforms both our runs on RMSE. On Pearson correlation, the baseline leads in German as well (0.773 vs. 0.772), while for Chinese our Run 2 reaches $r = 0.777$ against the baseline’s 0.753. The primary metric goes to the baseline on both languages.

The inconsistency across languages also points to a deeper issue: the hyperparameters that work for Spanish do not generalise. Spanish benefits from dense coverage in XLM-RoBERTa’s pretraining data and relatively transparent morphology; German and Chinese present harder targets for the same configuration.

5 Conclusion and Future Work

The system is straightforward: one pretrained encoder, per-language fine-tuning, a flat input string, two runs averaged together. The system makes no architectural contributions. What the results show is that this setup is sufficient for Spanish—RMSE = 1.170 and Pearson Correlation = 0.812 without external resources, language-specific features, or architectural modifications—but does not gener-

alise cleanly across languages.

The English-side signal warrants further investigation. The clue and target words may provide useful cross-lingual anchors, particularly for Chinese where the encoder’s L1 coverage is weakest, and future work should include a controlled ablation to isolate their contribution.

The clearest lesson from the results is that a single hyperparameter configuration cannot serve three typologically different languages equally well. Spanish, German, and Chinese respond differently to learning rate, batch size, and scheduler choice. Future work should tune these per language from the start, add explicit regularisation to the regression head, and retain per-run scores across all languages to enable proper ablation.

In the future, we also plan to adopt the use of LLMs, such as GPT, Claude, etc. for improving the performance of Lexical Complexity Prediction. While the usage of LLMs has been shown to perform quite well in tasks such as generating simplifications across multiple languages ((Aumiller and Gertz, 2022; Dutilleul et al., 2024)), to the best of our knowledge, not much work has gone into measuring lexical complexity across multiple languages.

Limitations

Intermediate per-run predictions were not saved for Spanish and Chinese, preventing us from constructing and evaluating ensemble variants for those languages post-hoc. The ablations mentioned in Section 3 were run on the validation set only and were not systematic; we adjusted one variable at a time without a full factorial search. The ensemble weights (0.6/0.4) were set by inspection across all languages rather than optimized per language—the German results suggest this was suboptimal.

Finally, the system uses no morphological analysis, frequency information, or psycholinguistic norms—features that have historically been strong predictors of lexical complexity and that might close some of the remaining gap.

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.
- 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. Association for Computational Linguistics.
- Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.
- Benjamin Dutilleul, Mathis Debaillon, and Sandeep Mathias. 2024. [ISEP_Presidency_University at MLSP 2024 shared task: 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)*, pages 605–609, Mexico City, Mexico. Association for Computational Linguistics.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6402–6413.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. [Multi-task deep neural networks for natural language understanding](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496. Association for Computational Linguistics.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. [On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines](#). In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.
- Sian North, Fernando Alva-Manchego, and Matthew Shardlow. 2022. Predicting multilingual sentence readability with transformer models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3054–3065. International Committee on Computational Linguistics.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo 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. Association for Computational Linguistics.
- Matthew Shardlow, Marcos Zampieri, and Richard Evans. 2022. [Predicting the complexity of words in context with lexical resources and transformer models](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 91–100. Association for Computational Linguistics.
- George-Eduard Zaharia, Radu Ion, and Dan Tufis. 2022. [Exploring transformer models for lexical complexity prediction](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 11–20. Association for Computational Linguistics.

TOEBM at BEA 2026 Shared Task 1: Improving Lexical Difficulty Prediction with Context-Aligned Contrastive Learning and Ridge Ensembling

Wicaksono Leksono Muhamad^{†,1}, Joanito Agili Lopo^{†,1}, Tsamarah Rana Nugraha^{♡,1,2},
Ahmad Cahyono Adi^{♡,1,3}, Muhammad Oriza Nurfajri³

¹Mantera Studio ²The University of Manchester ³Universitas Gadjah Mada
{wcksnlxn, amalopo99, ahmadseverine83}@gmail.com
tsamarah.nugraha@postgrad.manchester.ac.uk
oriza_nurfajri@mail.ugm.ac.id

Abstract

Lexical difficulty prediction is a fundamental problem in language learning and readability assessment, requiring models to estimate word difficulty across different first-language (L1) backgrounds. However, existing approaches rely on regression-only training with scalar supervision, which does not explicitly structure the representation space, limiting their ability to capture cross-lingual alignment and ordinal difficulty. To mitigate these issues, we propose *Context-Aligned Contrastive Regression*, which integrates Ridge regression ensemble with two complementary objectives, i.e., Cross-View Context and Ordinal Soft Contrastive Learning. Experiments on three L1 datasets show that (i) contrastive objectives improve cross-lingual representation alignment while preserving language-specific nuances, (ii) the learned representations capture the ordinal structure of lexical difficulty, and (iii) the ensemble effectively mitigates systematic biases of individual models, leading to more stable performance across difficulty levels.¹

1 Introduction

Vocabulary is a central component of English as a Foreign Language proficiency, supporting the development of reading, listening, writing, and speaking skills (Alshumrani, 2024). Since learners' proficiency levels influence how vocabulary is acquired (Bao and Peng, 2024), estimating lexical difficulty is an important step in the development of level-appropriate learning materials and valid assessment instruments (Goyibova et al., 2025)

Prior work has explored related tasks such as Complex Word Identification, Lexical Difficulty Prediction, and Lexical Simplification (Paetzold

and Specia, 2016b; Yimam et al., 2018; Shardlow et al., 2021). However, they were not tailored for English language learners and ignored how learners' first language (L1) can make English vocabulary easier or harder. This limitation is critical, as vocabulary knowledge is inherently multi-layered and involves several interrelated components that must be acquired for effective language use (Schmitt, 2010).

Among the interrelated components, L1 interference plays a central role across linguistic levels, including grammar, syntax, phonology, and vocabulary, leading to systematic differences in how learners from different L1 backgrounds perceive and process words (Alisoy, 2024). Consequently, lexical difficulty is not an intrinsic property of a word, but a relational phenomenon that varies across L1 backgrounds, motivating the need for L1-aware modeling (Skidmore et al., 2025).

However, learner-specific factors alone are insufficient. The ability to distinguish similar-sounding words does not guarantee successful acquisition or comprehension (Pajak et al., 2016). Therefore, lexical difficulty cannot be reliably inferred from form-level properties alone. Instead, difficulty is shaped by how words are encountered and interpreted in context. Psycholinguistic evidence shows that contextual cues guide meaning interpretation and comprehension (Garten et al., 2019). Effective lexical difficulty modeling should therefore capture both learner-specific and contextual dimensions.

To address these challenges, we propose *Context-Aligned Contrastive Regression*, a multi-objective framework for L1-aware lexical difficulty prediction. Our approach (i) integrates direct regression with Cross-View Context Contrastive Learning to align representations across lexical views, (ii) incorporates Ordinal Soft Contrastive Learning to preserve graded difficulty structure, and (iii) leverages complementary encoder models through Ridge ensembling. Together, these components enable more

[†] Main contributors.

[♡] Major contributors.

¹<https://github.com/airlangawicaksono/BEA2026TOEBM>

robust, aligned, and interpretable difficulty estimation.

2 Background

Recent studies show that transformer-based models can predict lexical difficulty from contextualized and multilingual representations (Shardlow et al., 2021, 2024; Skidmore et al., 2025). These approaches are motivated by the growing importance of lexical complexity prediction in applications such as text simplification, readability assessment, and language learning (Shardlow, 2022; Rotaru, 2021). In particular, shared tasks such as SemEval-2021 have demonstrated that fine-tuned transformer models can achieve strong performance by leveraging contextual information (Shardlow et al., 2021; Rotaru, 2021).

However, most approaches rely on regression-only training, where supervision is given only through a scalar difficulty score. This can improve prediction accuracy, but it does not directly structure the representation space. As a result, items with similar difficulty may not be close in the latent space, while items with different difficulty levels may not be clearly separated. This limitation becomes more pronounced in multilingual settings, where models must capture both cross-lingual alignment and language-specific variation (Skidmore et al., 2025).

Contrastive learning addresses this limitation by shaping representations through relations between examples (Chen et al., 2020; Khosla et al., 2020). This fits lexical difficulty prediction, where difficulty depends on the target word, context, and learner-specific variation. Recent contrastive regression methods define similarity by proximity in continuous target values rather than discrete labels (Zha et al., 2023; Keramati et al., 2024), making them suitable for ordinal difficulty scores. We therefore formulate L1-aware lexical difficulty prediction as a multi-objective problem that combines direct regression with auxiliary contrastive supervision.

3 Context-Aligned Contrastive Regression

Building on recent work in lexical difficulty prediction and L1-aware modeling (Shardlow et al., 2021, 2024; Skidmore et al., 2025), we propose *Context-Aligned Contrastive Regression*, which integrates direct regression with representation-level

contrastive regularization. Given an input instance enriched with L1-aware contextual information, such as translated context and English target information, the model encodes multiple contextual views into a shared representation space. The resulting representation is used for both difficulty prediction and contrastive learning.

Specifically, we optimize the model using Regression loss (§3.1) for direct difficulty prediction, a Cross-View Context Contrastive loss (§3.2) for learning view-invariant contextual representations, and an Ordinal Soft Contrastive loss (§3.3) for encoding the continuous ordering of difficulty scores in the latent space. By combining this, the overall training objective does not merely fit the target score, but also provides a representation space that is both contextually stable and smoothly aligned with the ordinal structure of the task. The general explanation of our system is presented in the Figure 1.

3.1 Regression Objective

The regression objective is used to supervise the final prediction directly. Let $h_i \in \mathbb{R}^d$ denote the shared encoder representation of item i , and let $\hat{y}_i$ denote the scalar prediction generated by the regression head. The model is optimized with the mean squared error:

$$\mathcal{L}_{\text{reg}} = \frac{1}{B} \sum_{i=1}^B (\hat{y}_i - y_i)^2, \quad (1)$$

where y_i is the label associated with item i and B is the batch size.

While this objective provides direct supervision for score prediction, it does not explicitly enforce representation consistency across alternative contextual realizations of the same lexical item, nor does it preserve proximity structure among items with similar difficulty levels. To address these limitations, we introduce two auxiliary contrastive objectives that regularize the representation space with respect to contextual alignment and ordinal difficulty structure.

3.2 Cross-View Context Contrastive Objective

Lexical Difficulty is largely determined by intrinsic properties of the target word rather than by superficial variation in context (Paetzold and Specia, 2016a; Gooding and Kochmar, 2018; Shardlow et al., 2021). However, contextualized encoders can mix lexical information with context-specific cues,

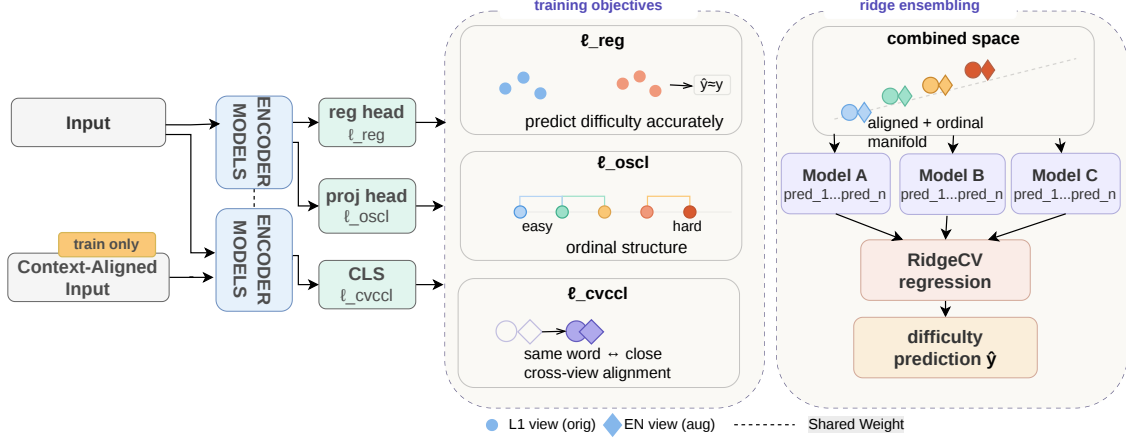


Figure 1: The proposed method combines regression and contrastive auxiliary objective, including cross-view alignment and ordinal-aware representation learning, to learn representations that are aligned across views and difficulty levels. Outputs from multiple encoder models are ensembled using ridge regression to produce the final prediction.

which may produce inconsistent representations for the same lexical item across different contexts.

To reduce this effect, we use a cross-view contrastive objective for lexical difficulty prediction. Unlike standard contrastive learning methods that rely on stochastic augmentations (van den Oord et al., 2019; Chen et al., 2020), our method uses task-specific paired views derived from the L1-aware input and the context-aligned representation described in Section 4.2. This encourages the encoder to learn representations that remain stable across contextual variation while still distinguishing different lexical items (Khosla et al., 2020).

$$z_i^{\text{tgt}} = H(en_tgt_i) \quad (2)$$

$$z_i^{\text{full}} = H(full_input_i) \quad (3)$$

Given a mini-batch of size B , each instance is encoded through two matched views, en_tgt (Eq. 2) and $full_input$ (Eq. 3), yielding $2B$ representations. Since lexical difficulty prediction is formulated as a regression task, positive pairs are defined by cross-view correspondence rather than shared class labels. Thus, each anchor representation z_i is paired with its matched representation from the alternative view. The objective is formalized as

$$\mathcal{L}_{CVCL} = -\frac{1}{2B} \sum_{i=1}^{2B} \log \frac{\exp(z_i^T z_{i+}/\tau)}{\sum_{k=1}^{2B} \mathbf{1}_{[k \neq i]} \exp(z_i^T z_k/\tau)} \quad (4)$$

3.3 Ordinal Soft Contrastive Objective

While the cross-view objective promotes consistency across views, it does not explicitly capture the ordinal structure of lexical difficulty. Since difficulty is represented as a continuous psychometric estimate rather than a discrete class label (Shardlow et al., 2021), items with nearby scores should be closer in representation space than items with distant scores. Standard contrastive objectives rely on instance discrimination or discrete class supervision (Chen et al., 2020; Khosla et al., 2020), while recent regression-oriented contrastive methods address this limitation by organizing representations according to target distance or order (Zha et al., 2023; Keramati et al., 2024; Xue et al., 2024).

Following this motivation, we introduce an ordinal soft contrastive objective that replaces binary pair assignments with continuous pairwise weights derived from score proximity. This allows the model to preserve the graded structure of lexical difficulty in the embedding space. Let u_i denote the representation of item i , and let y_i be its lexical difficulty score. For a batch of size B , we define the affinity between items i and j as

$$w_{ij} = \exp\left(-\frac{(y_i - y_j)^2}{2\sigma^2}\right), \quad w_{ii} = 0, \quad (5)$$

where σ controls how strongly the objective responds to differences in lexical difficulty. Pairs with similar scores receive larger weights, while pairs with distant scores receive smaller weights.

We then define the similarity distribution for anchor u_i over the remaining items in the batch as

$$p_{ij} = \frac{\exp(\text{sim}(u_i, u_j)/\tau)}{\sum_{k \neq i} \exp(\text{sim}(u_i, u_k)/\tau)}, \quad j \neq i, \quad (6)$$

where $\text{sim}(u_i, u_j)$ denotes cosine similarity and τ is a temperature parameter. The ordinal soft contrastive loss is then defined as

$$\mathcal{L}_{\text{OSCL}} = \frac{1}{B} \sum_{i=1}^B \left(-\frac{\sum_{j \neq i} w_{ij} \log p_{ij}}{\sum_{j \neq i} w_{ij}} \right). \quad (7)$$

This objective complements the cross-view contrastive loss by shaping the representation space according to graded difficulty similarity. Items with nearby difficulty scores are encouraged to lie closer together, while items with larger score differences exert weaker attractive force. As a result, the learned embedding space reflects the continuous and ordinal nature of lexical difficulty more faithfully.

3.4 Ridge-Based Ensemble

After training the model with the proposed multi-objective learning framework, we further enhance prediction performance by leveraging an ensemble of models with different encoder backbones. Each model is trained independently using the same objective, resulting in diverse yet complementary predictions.

Let $f_k(x)$ denote the prediction of the k -th model for an input x . We construct a meta-representation by stacking predictions from K models:

$$\mathbf{z}(x) = [f_1(x), f_2(x), \dots, f_K(x)]. \quad (8)$$

We then apply ridge regression to combine these predictions, allowing the model to learn adaptive weights over individual predictors while maintaining robustness through regularization. This approach provides a more flexible alternative to simple averaging and leads to more stable and accurate predictions.

4 Experimental Setup

4.1 Dataset

We perform lexical difficulty prediction as a regression task using the multilingual L1-aware dataset introduced by Skidmore et al. (2025). The data are

organized into three first-language groups, namely German, Spanish, and Mandarin Chinese. Total of each L1 group is presented in Table 1. For the translated training and development splits, we additionally include *en_context*, as described in Section 4.2.

Furthermore, the target variable, *GLMM_score*, represents the estimated difficulty of each vocabulary item. It is computed from large-scale learner response data using a generalized linear mixed model (GLMM) framework, which models both item-level and learner-level variability (Schmitt et al., 2024). Lower scores indicate greater lexical difficulty.

Table 1: Data distribution across L1 groups and splits.

L1 group	Train	Dev	Test
German	6,091	677	748
Spanish	6,091	677	748
Mandarin Chinese	6,091	677	748
Total	18,273	2,031	2,244

4.2 L1-Aware Input Representation

Skidmore et al. (2025) reported that their best-performing setup used a multilingual model trained jointly on all L1 subsets of the KVL together with an L1-aware input representation. Following this formulation, we represent each instance as a single sequence that combines the L1 source word (w), its L1 context (ctx), the English clue ($clue$), and the target English word (tgt). These components are separated by a special token before being passed to a pretrained encoder, as shown in the example input text below:

casa [SEP] Vivo en una casa grande
que tiene tres dormitorios. [SEP]
h_____ [SEP] house

In addition, we construct a context-aligned view by combining the English target word from the source language input (*en_tgt*) with the translated English context (*en_ctx*), which is obtained by translating the source-language context *ctx*.² This augmented view is then used to form contrastive pairs for cvcl objective (§3.2).

²We use google/mt5-large to translate the source-language context.

4.3 Model Architecture

Our model follows a shared-encoder architecture with task-specific heads for regression and contrastive learning. Given an input sequence, the encoder produces a contextualized representation from the [CLS] token, which is then used for both prediction and representation-level objectives.

Projection Head For contrastive learning, we apply a projection head that maps the encoder representation into a lower-dimensional space. The projected representation is used for contrastive objectives. Separating the projection space from the original representation allows the model to preserve task-relevant features for regression while learning more structured representations for contrastive objectives.

Models Setting We use three multilingual encoder models as base learners in the ensemble: XLM-RoBERTa (Conneau et al., 2020), multilingual DeBERTaV3 (He et al., 2023), and mmBERT (Marone et al., 2025). At inference time, predictions from all base learners are combined using a ridge regression meta-model, which learns to aggregate their outputs into a final prediction. The hyperparameters and training settings are reported in Appendix D.

5 Results & Discussion

In this section, we present our main result (see Table 2 and Appendix A for detailed results) along with its observations and discussion. We also provide qualitative examples of analysis across languages, describing the top-5 improved and failure cases in the dataset.

5.1 Main Result

Performance Across Languages Table 2 shows that the ensemble consistently outperforms individual base models across the three L1 groups. These gains suggest that combining input design, auxiliary objectives, and model ensembling provides more robust predictions than relying on a single encoder. Spanish achieves the best performance with the *en_ctx* setting, while German and Chinese benefit more from *en_tgt*-based representations. Rather than indicating instability, this pattern suggests that lexical difficulty is shaped by different linguistic cues across L1 backgrounds. This makes the variation between input settings analytically

meaningful and motivates a deeper examination of model behavior and representation structure.³

Effect of Input and Objective Design A closer comparison of input and objective variants shows that more complex configurations do not always improve performance. Although *full_input* (§4.2) and *en_ctx* + *en_tgt* perform strongly, they do not consistently outperform focused single-view representations, such as *en_tgt* (CVCCCL + OSCL) or *en_context* (CVCCCL + OSCL). This suggests that carefully selected inputs can be more effective than combining all available features, likely by reducing noise and improving representation stability.

At the objective level, CVCCCL + OSCL remains consistently competitive across languages. It does not always achieve the best score, but it stays among the top configurations in each setting. This indicates that contrastive alignment and ordinal supervision provide complementary learning signals, while their effectiveness depends on the structure of the input information.

BEA 2026 Submission We submitted our best model to the BEA 2026 Shared Task Closed Track. Full leaderboard comparisons are provided in Appendix A. Our submission, **TOEBM**, achieved top-15 performance across all L1 groups, ranking 14th for Spanish, 11th for German, and 7th for Mandarin Chinese, as shown in Tables 5, 6, and 7, respectively. This pattern suggests that the proposed approach remains reasonably robust across different L1 backgrounds, even when the relative difficulty cues vary by language. Although its RMSE is about ± 0.125 higher than the top-ranked Glite Team on average, the model maintains competitive Pearson correlation scores across languages, with 0.832 for Spanish and German, and 0.853 for Mandarin Chinese. These results indicate that, while the model is not yet fully optimized for minimizing prediction error, it captures the relative ordering of lexical difficulty effectively.

5.2 Representation Analysis

To better understand the impact of the contrastive objectives, we analyze the learned representations beyond the final prediction layer. In particular, we examine whether the contrastive losses improve cross-lingual alignment (§3.2) and preserve the ordinal structure of lexical difficulty (§3.3) across

³The Ensemble Baseline combines the XLM-R, mDeBERTa-v3, and mmBERT baseline models using ridge regression.

Model	ES			DE			CN		
	RMSE ↓	MSE ↓	ρ ↑	RMSE ↓	MSE ↓	ρ ↑	RMSE ↓	MSE ↓	ρ ↑
Individual Model									
XLM-R (base)	1.257	1.580	0.765	1.258	1.583	0.773	1.140	1.300	0.753
mDeBERTa-v3 (base)	1.151	1.326	0.836	1.160	1.344	0.833	1.027	1.054	0.855
mmBERT (base)	1.079	1.164	0.825	1.010	1.021	0.826	0.911	0.830	0.843
Ensemble Variants									
ensemble baseline	1.051	1.104	0.835	0.998	0.996	0.831	0.893	0.798	0.848
full_input (CVCCL + OSCL)	1.045	1.092	0.835	1.016	1.032	0.828	0.907	0.823	0.843
en_ctx + en_tgt (CVCCL + OSCL)	1.052	1.107	0.831	1.004	1.008	0.829	0.892	0.796	0.846
en_ctx (CVCCL + OSCL)	1.041	1.084	0.837	1.010	1.020	0.827	0.890	0.792	0.852
en_tgt (CVCCL + OSCL)	1.063	1.130	0.826	0.997	0.994	0.832	0.880	0.774	0.853
en_tgt (CVCCL)	1.052	1.107	0.831	0.998	0.996	0.831	0.891	0.794	0.847
en_tgt (OSCL)	1.077	1.160	0.825	1.001	1.002	0.832	0.866	0.750	0.858

Table 2: Performance comparison with RMSE, MSE, and Spearman’s ρ across languages and models in the BEA shared task test set. Individual model denotes fine-tuned model with regression head on top of the architecture and *full_input* or L1-aware input representation §4.2 as model input. Lower RMSE/MSE and higher ρ indicate better performance.

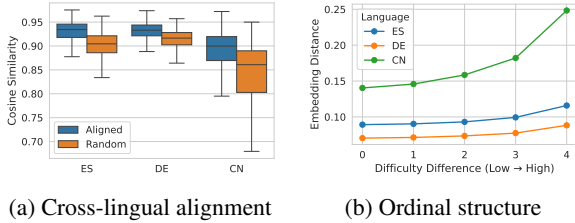


Figure 2: Representation analysis across Spanish, German, and Chinese using mmBERT-base. (a) Cosine similarity between aligned cross-lingual lexical pairs and randomly paired inputs. (b) Relationship between embedding distance and absolute lexical difficulty score differences, larger score gaps correspond to greater representational separation.

multiple languages.

Cross-lingual Representation Alignment We first examine whether the contrastive objective leads to improved cross-lingual alignment in the learned representations. To this end, we measure cosine similarity between representations of semantically equivalent inputs (L1 and English) and compare them against randomly paired inputs.⁴ As shown in Figure 2a, aligned pairs consistently exhibit higher similarity than random pairs across all three languages (ES, DE, and CN), indicating that the model learns a shared semantic space across languages.

Notably, the degree of separation varies across languages, with German showing the clearest gap, while Chinese exhibits greater variance and over-

lap. The presence of overlap between the two distributions suggests that the alignment is not fully separable. This indicates that while the contrastive objective contributes to reducing cross-lingual representation gaps, it does not enforce strict alignment. We hypothesize that this moderate alignment is beneficial, as it preserves language-specific nuances that may still be relevant for lexical difficulty prediction.

Ordinal Structure in Representation We next analyze whether the learned representations capture the ordinal nature of lexical difficulty. Specifically, we examine the relationship between embedding distances and differences in GLMM scores⁵. As shown in Figure 2b, we observe a clear and consistent increasing trend across all three languages, where pairs of instances with larger difficulty differences exhibit greater distances in the embedding space.

We further quantify this trend using Spearman’s rank correlation. We observe a consistent positive monotonic relationship across all three languages (Spearman’s $\rho = 0.22$ for ES, 0.20 for DE, and 0.39 for CN), indicating that embedding distances tend to increase with larger differences in lexical difficulty. While the correlation is moderate in magnitude, its consistency across languages suggests that the ordinal structure is reliably encoded in the learned representations, although the relationship is not strictly linear.

⁴The details of measurement technique is presented in Appendix F.

⁵The details of measurement technique is presented in Appendix E

Condition	Low	High	Rel. ↓ (%)
Context Length	0.841	0.834	0.8
Lexical Diversity	0.847	0.837	1.2
Target Word Length	0.843	0.831	1.4
Orthographic Complexity	0.855	0.813	4.9

Table 3: Mean absolute error (MAE) of the Ridge ensemble across input characteristics on the Spanish dataset. Relative improvement (%) measures the reduction from low to high condition. Lower values indicate better performance.

Input Representation Behavior Beyond representation structure, we further investigate how input characteristics influence prediction behavior. Specifically, we analyze prediction error with respect to several surface-level input properties, including context length, lexical diversity (measured via type-token ratio), target word length, and orthographic complexity. We observe a generally consistent pattern in which richer contextual and lexical signals are associated with lower prediction error. In particular, longer and lexically more diverse contexts, as well as more orthographically informative target words, tend to improve prediction quality, suggesting that contextual and lexical richness provide useful signals for lexical difficulty modeling.

However, the strength of this effect varies across languages. As shown in Table 3, for Spanish, the improvements are more pronounced, especially in lexically diverse contexts and orthographically complex words, which exhibit the largest relative gain. This indicates that Spanish benefits more from contextualized representations, explaining why the *en_ctx* configuration outperforms target-only inputs in this setting. In contrast, for other languages, the gains from context are less dominant, suggesting that target-side information can be sufficient depending on the dataset characteristics. This finding is consistent with the representation-level analysis, where contextual signals contribute to richer and more structured embeddings, which in turn improve downstream prediction.

5.3 Ensemble Investigation

We analyze the contribution of each base model to the meta-input of the Ridge regression ensemble. This analysis aims to assess how well individual models approximate the GLMM scores and whether they provide complementary prediction patterns that can be effectively leveraged by the ensemble.

Base Models Prediction Behaviour Figure 3 illustrates the relationship between predicted scores and ground truth GLMM scores across all three L1 groups. The models exhibit systematic differences in how lexical difficulty is estimated. XLM-RoBERTa consistently produces a relatively flat trend, particularly in the Chinese dataset, indicating a tendency to compress predictions toward the mean and underestimate more difficult items. In contrast, mDeBERTa-v3 shows a steeper slope, often overshooting the ground-truth line, suggesting systematic overestimation for complex lexical items. Meanwhile, mmBERT demonstrates a more moderate slope that more closely follows the optimal diagonal trend, indicating a more balanced estimation behaviour.

Statistical Patterns These contrasting trends reveal that the base models capture different aspects of the lexical difficulty space. While XLM-RoBERTa tends to under-predict and mDeBERTa-v3 tends to over-predict, mmBERT produces more balanced predictions with errors closer to zero and generally competitive variance. This complementary behavior is further reflected in the trade-off between correlation and error metrics, where mDeBERTa-v3 often achieves higher Pearson correlation while mmBERT attains lower RMSE and MAE in several cases.

When considered alongside the statistical results in Table 8 (Appendix), these findings suggest that the models are not redundant but instead provide diverse and complementary signals. Such diversity is crucial for the effectiveness of the ensemble, as it enables the meta-learner to reconcile systematic biases and produce more accurate final predictions.

Difficulty-wise Error Analysis To further evaluate ensemble effectiveness across the lexical difficulty spectrum, we analyze model errors over five GLMM-based difficulty bins averaged across all three L1 groups. Figure 4 presents the resulting mean absolute error (MAE) patterns.

We observe that XLM-RoBERTa exhibits a substantial increase in error from harder to easier lexical items, indicating that its predictions become progressively less reliable for comparatively easier words and suggesting strong score compression effects. Similarly, mmBERT demonstrates moderate but consistent error growth toward easier bins, although its overall performance remains more balanced than XLM-RoBERTa. In contrast, mDeBERTa-v3 maintains relatively stable perfor-

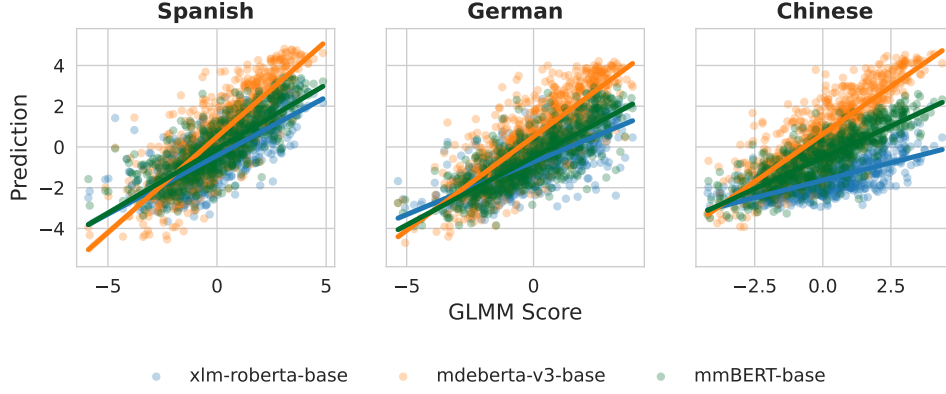


Figure 3: Relationship between predicted scores and ground-truth GLMM scores across Spanish (ES), German (DE), and Chinese (CN). Each subplot shows regression trends for individual base models, highlighting systematic differences in prediction behavior, including underestimation, overestimation, and balanced estimation patterns.

mance across the full spectrum, with only minor variation between difficulty levels.

Importantly, the Ridge ensemble consistently achieves the lowest or near-lowest MAE across nearly all bins while maintaining the flattest error profile overall. This suggests that the meta-learner effectively reconciles the complementary weaknesses of individual base models, substantially reducing systematic prediction bias and producing more stable lexical difficulty estimation across both harder and easier lexical items.

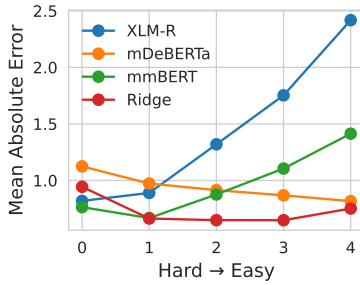


Figure 4: Mean absolute error (MAE) across lexical difficulty bins, averaged over all three L1 groups. Difficulty bins are derived from GLMM score quantiles and ordered from harder (lower GLMM scores) to easier (higher GLMM scores) lexical items.

5.4 Improved vs Failures Cases

To complement the quantitative findings in Section §5.1, we further examine representative cases to understand how the ensemble model corrects prediction errors from individual base models.

Error Correction Table 4 presents selected examples across Spanish, Chinese, and German. We observe that the ensemble consistently corrects

Lang	Word	GLMM	mmBERT	Ridge
ES	cruise	1.21	-0.50	1.23
ES	run	2.20	0.31	1.99
ES	character	1.08	-1.11	0.55
ES	dining	-4.67	-0.76	0.19
ES	baking	-3.78	-0.20	0.87
CN	schoolmate	1.32	-2.55	-0.09
CN	dating	0.64	-2.31	-0.001
CN	goldfish	1.51	-1.44	0.63
CN	double	-2.58	0.11	0.82
CN	stake	-4.19	-1.67	-1.13
DE	cruise	1.21	-1.48	0.60
DE	handbook	0.95	-1.36	0.31
DE	workbook	2.10	-2.07	-0.44
DE	anger	-3.39	0.10	0.57
DE	umbrella	3.24	-1.13	-0.20

Table 4: Qualitative examples across Spanish (ES), Chinese (CN), and German (DE), comparing mmBERT-base and Ridge ensemble predictions. The first three rows per language show cases where the ensemble corrects large underestimation errors from base models, while the last two rows highlight failure cases.

large underestimation errors from individual models. For instance, words such as *cruise*, *run*, and *character* in Spanish, as well as *schoolmate* and *dating* in Chinese, are substantially underestimated by mmBERT, yet the Ridge ensemble successfully adjusts their predictions toward the ground-truth GLMM scores. Similar correction patterns are observed in German, suggesting that the ensemble effectively leverages complementary signals across models to reduce systematic bias.

Failure Cases Despite these substantial improvements, certain lexical items remain challenging even after ensembling. In several cases, such as *dining* and *baking* in Spanish, *double* and *stake* in

Chinese, and *anger* and *umbrella* in German, the predicted difficulty remains far from the ground-truth values. These errors often involve strong overestimation or residual bias, indicating that certain lexical items remain challenging due to semantic ambiguity or context-dependent interpretations. This suggests that while the ensemble mitigates systematic biases, it cannot fully resolve cases where all base models fail to capture the underlying difficulty signal.

6 Conclusion

Our method consistently outperforms individual multilingual encoders across Spanish, German, and Chinese learner groups, effectively correcting systematic under- and overestimation biases while maintaining stable performance across all difficulty levels. Representation analysis confirms that the contrastive objectives enhance cross-lingual alignment and preserve the ordinal structure of lexical difficulty in the embedding space. Collectively, these results demonstrate that combining direct regression with representation-level regularization significantly improves both the accuracy and structural reliability of L1-aware lexical difficulty prediction.

Acknowledgments

The authors gratefully acknowledge Mantera Studio for providing the computational resources and APIs utilized in this research. Additionally, this work was partially supported by the 2026 Publication Funding from the Department of Computer Science and Electronics at Universitas Gadjah Mada.

Limitations

Although the proposed method improves performance across the evaluated L1 groups, several limitations remain. First, the experiments are limited to Spanish, German, and Mandarin Chinese, so the findings may not fully generalize to other L1 backgrounds. Second, the use of contrastive objectives introduces additional training objectives and weighting hyperparameters, which may increase sensitivity to objective balancing across languages. Third, the ridge ensemble improves predictive performance by combining multiple encoders, but it also increases computational cost because several models must be trained and evaluated. Future work should examine more robust weighting strategies, reduce the computational overhead of ensembling,

and evaluate the method on a wider range of L1 backgrounds.

Ethical Considerations

We acknowledge that our research utilized AI tools for writing, rewriting, and generating code. Although these tools offer significant advantages in terms of efficiency and productivity, their use raises important ethical considerations. We recognize the potential for bias and errors inherent in AI-generated content and have taken steps to mitigate these risks through rigorous human review and validation. Furthermore, we are mindful of the potential impact on the broader software development community, particularly regarding job displacement and the need for upskilling. We believe that responsible AI integration should prioritize transparency, accountability, and the empowerment of human developers, ensuring that these tools augment rather than replace human expertise. This research aims to contribute to the ongoing dialogue on ethical AI development and usage, advocating for a future where AI tools are harnessed responsibly to enhance human creativity and innovation in the field of software engineering.

References

- Hasan Alisoy. 2024. [Exploring language acquisition: The role of native language interference in esl learners](#). *Journal of Azerbaijan Language and Education Studies*, 1(1):50–66.
- H. A. Alshumrani. 2024. [Unveiling vocabulary teaching and learning beliefs of teachers and learners in an EFL context](#). *Asian Journal of Second and Foreign Language Education*, 9(1):20.
- Z. Bao and C. Peng. 2024. [The effects of EFL wordlist and proficiency on vocabulary knowledge](#). *Frontiers in Psychology*, 15:1289106.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. [A simple framework for contrastive learning of visual representations](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–

- 8451, Online. Association for Computational Linguistics.
- Justin Garten, Blair Kennedy, Kenji Sagae, and Morteza Dehghani. 2019. [Measuring the importance of context when modeling language comprehension](#). *Behavior Research Methods*, 51(2):480–492.
- 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.
- Nigora Goyibova, N. Muslimov, Barnokhon Samatova, and 1 others. 2025. [Differentiation approach in education: Tailoring instruction for diverse learner needs](#). *International Journal of Educational Research*, 125:102501. Accessed via ScienceDirect.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *Preprint*, arXiv:2111.09543.
- Mahsa Keramati, Lili Meng, and R. David Evans. 2024. [Conr: Contrastive regularizer for deep imbalanced regression](#). In *The Twelfth International Conference on Learning Representations*.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. [Supervised contrastive learning](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. [mmbert: A modern multilingual encoder with annealed language learning](#). *Preprint*, arXiv:2509.06888.
- Gustavo Paetzold and Lucia Specia. 2016a. [Inferring psycholinguistic properties of words](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 435–440, San Diego, California. Association for Computational Linguistics.
- Gustavo Paetzold and Lucia Specia. 2016b. [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.
- Bozena Pajak, Sarah C. Creel, and Roger Levy. 2016. [Difficulty in learning similar-sounding words: A developmental stage or a general property of learning?](#) *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(9):1377–1399.
- 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.
- N. Schmitt, K. Dunn, B. O’Sullivan, L. Anthony, and B. Kremmel. 2024. [Knowledge-based Vocabulary Lists](#). British Council Monographs on Modern Language Testing. University of Toronto Press.
- Norbert Schmitt. 2010. *Researching Vocabulary: A Vocabulary Research Manual*. Palgrave Macmillan, Basingstoke.
- Matthew Shardlow. 2022. [Agree to disagree: Exploring subjectivity in lexical complexity](#). In *Proceedings of the 2nd Workshop on Tools and Resources to Empower People with READING Difficulties (READI) within the 13th Language Resources and Evaluation Conference*, pages 9–16, Marseille, France. European Language Resources Association.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. [Representation learning with contrastive predictive coding](#). *Preprint*, arXiv:1807.03748.
- Tengfei Xue, Fan Zhang, Leo R. Zekelman, Chaoyi Zhang, Yuqian Chen, Suheyly Cetin-Karayumak, Steve Pieper, William M. Wells, Yogesh Rathi, Nikos Makris, Weidong Cai, and Lauren J. O’Donnell. 2024. [Tractoscr: a novel supervised contrastive regression framework for prediction of neurocognitive measures using multi-site harmonized diffusion mri tractography](#). *Frontiers in Neuroscience*, 18:1411797.

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.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. 2023. [Rank-n-contrast: Learning continuous representations for regression](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.

A BEA 2026 Leaderboard Comparison

Table 5, Table 6, and Table 7 present selected leaderboard comparisons for the Spanish, German, and Chinese subtasks, respectively. Rather than listing the complete leaderboard, we report the highest-ranked systems and nearby reference systems to contextualize our submission. Our team is registered as **TOEBM**. In the closed-track results, TOEBM ranked 14th for Spanish, 11th for German, and 7th for Chinese, with RMSE scores of 1.063, 0.997, and 0.880, respectively. The leaderboard scores differ slightly from those reported in the main experimental section because the present paper uses a unified prompt setting across all subtasks, while the official leaderboard reflects the submitted prediction files. Nevertheless, the relative ranking of our submission remains unchanged.

#	Team Name	Prediction Type	RMSE	Pearson
1	Glite	predictions_run_3	0.903	0.877
...	...	...	...	...
7	Sakura	predictions_closed_max	0.983	0.854
11	NLP-Explorers	predictions_run_3	1.041	0.838
...	...	...	...	...
14	TOEBM	predictions_run_3	1.063	0.826

Table 5: Leaderboard comparison of Closed Track Spanish Language

#	Team Name	Prediction Type	RMSE	Pearson
1	Glite	predictions_run_3	0.885	0.871
...	...	...	...	...
4	Uogal	8enc_dmeta_elasticnet...csv	0.903	0.869
7	NLP-Explorers	predictions_run_3	0.992	0.845
...	...	...	...	...
11	TOEBM	predictions_run_3	0.997	0.832

Table 6: Leaderboard comparison of Closed Track German Language

#	Team Name	Prediction Type	RMSE	Pearson
1	Glite	predictions_run_3	0.776	0.889
...	...	...	...	...
4	Sakura	predictions_closed_max	0.816	0.874
5	uogal	8enc_dmeta_elasticnet...csv	0.820	0.879
...	...	...	...	...
7	TOEBM	predictions_run_3	0.880	0.853

Table 7: Leaderboard comparison of Closed Track Chinese Language

B Base Model Statistic

C Representation analysis

To analyze the learned representation space, we distinguish between the Ridge ensemble used for prediction and the fused embedding used for visualization. The Ridge ensemble prediction is defined as

$$\hat{y}(x) = b + \sum_{n=1}^N w_n f_n(x), \quad (9)$$

where $f_n(x)$ denotes the scalar prediction produced by the n -th base model, w_n is the learned Ridge coefficient for that model, and b is the intercept. This formulation is used only for final score prediction. It is not suitable for direct t-SNE visualization because it produces a one-dimensional output.

For representation analysis, we instead construct a fused embedding by concatenating the weighted embedding vectors from each base model:

$$z(x) = [w_1 E_1(x); w_2 E_2(x); \dots; w_N E_N(x)], \quad (10)$$

where $E_n(x) \in \mathbb{R}^{d_n}$ is the embedding produced by the n -th encoder. In this study, $E_n(x)$ is obtained from the CVCCL+OSCL model using the *en_tgt* view as the encoder input. The Ridge coefficients are then used as model-level weights to scale each embedding before concatenation.

We apply t-SNE to the fused representation $z(x)$ to visualize the structure of the embedding space. The resulting points are grouped into five difficulty bins based on the minimum and maximum GLMM scores. This allows us to inspect whether examples with similar lexical difficulty are located close to one another in the representation space, while the Ridge ensemble prediction $\hat{y}(x)$ is used separately to evaluate predictive performance. Figure 5 shows that the separation across difficulty bins is generally strong, although the quality of the structure varies across models and languages. Since GLMM is a continuous score and the bins are created only for visualization, perfect discrete clusters are not

Metrics	Español			Deutsch			Chinese		
	XLM-R	mDeBERTa	mmBERT	XLM-R	mDeBERTa	mmBERT	XLM-R	mDeBERTa	mmBERT
RMSE	1.326	1.266	1.113	1.441	1.219	1.154	2.070	1.135	1.091
MAE	1.066	0.993	0.861	1.145	0.959	0.908	1.793	0.891	0.880
Pearson	0.746	0.836	0.808	0.719	0.832	0.813	0.689	0.858	0.824
Mean Error	-0.432	0.497	-0.098	-0.722	0.507	-0.491	-1.644	0.628	-0.520
Std Error	1.253	1.165	1.109	1.247	1.108	1.045	1.258	0.945	0.960

Table 8: Performance and error analysis of base models across three L1 groups. Best values per language are highlighted in bold.

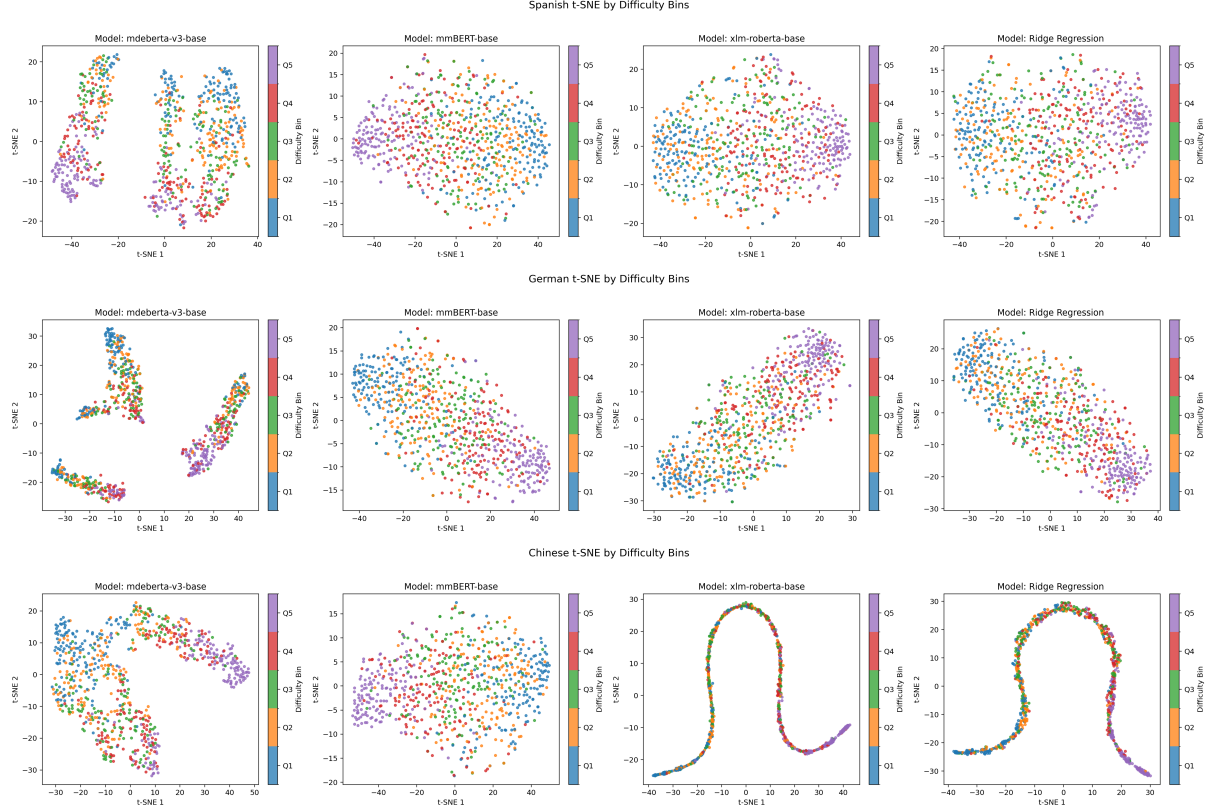


Figure 5: t-SNE visualization of fused model representations for Spanish, German, and Chinese. Each point represents one sample, and colors indicate five GLMM difficulty bins constructed from the minimum and maximum GLMM scores. The visualization is based on the weighted concatenation of encoder embeddings, while $\hat{y}(x)$ is used separately for predictive evaluation.

expected. Even so, a clear ordinal organization is visible in many panels, where neighboring regions are dominated by adjacent difficulty bins and the progression from lower to higher difficulty appears smooth.

Among the individual encoders, mdeberta-v3-base often produces well-separated local clusters, especially in Spanish and German, indicating that the learned space captures meaningful difficulty structure. mmBERT-base also shows some ordering, but the bins are more mixed and the boundaries between difficulty levels are less distinct. In contrast, xlm-roberta-base provides the clearest structure for Mandarin Chinese. In

the Chinese panel, the bins are arranged along a highly organized manifold with a very strong gradual transition from easier to harder examples. This suggests that xlm-roberta-base captures the ordinal nature of lexical difficulty particularly well for Mandarin Chinese.

The Ridge-weighted fused representation remains competitive and in some cases preserves a smooth global arrangement of the bins, showing that the ensemble combines complementary information from the base encoders. However, the clearest single-model structure for Mandarin Chinese is produced by xlm-roberta-base. Overall, these visualizations support the claim that the

CVCCCL+OSCL representations encode lexical difficulty in a meaningful way, with strong separation between bins and especially clear ordinal structure in the Mandarin Chinese setting.

D Training Settings

All models are trained using the Adam optimizer for 5 epochs with a learning rate of $2e-5$, a batch size of 8, weight decay of 0.01 , and a warmup ratio of 0.06 . The best model is selected based on the Mean Squared Error (MSE) on the validation set. The maximum input length is set to 128 tokens, while the maximum length for `en_context` is limited to 32 tokens. Furthermore, we set $\lambda_{cv} = 0.1$ and $\lambda_{ord} = 0.2$. The temperature parameter τ is set to 0.01 and 0.02 for the `cvcc1` and `oscl` losses, respectively.

For the Ridge regression meta-model, we consider a set of candidate regularization parameters $\alpha \in \{0.01, 0.1, 1.0, 10.0, 100.0\}$ and select the best value via cross-validation. Finally, We use Root Mean Squared Error (RMSE) as the primary metric for system ranking and report Pearson correlation for completeness.

E Ordinal Structure Analysis Details

To quantitatively assess whether the learned representations preserve the ordinal structure of lexical difficulty, we analyze the relationship between embedding distances and differences in ground-truth difficulty scores. Given a set of instances $\{(x_i, y_i)\}_{i=1}^N$, where y_i denotes the GLMM difficulty score, we first obtain normalized sentence embeddings $\mathbf{h}_i \in \mathbb{R}^d$. For a sampled subset of size N' , we compute all pairwise cosine distances:

$$d_{ij} = 1 - \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \|\mathbf{h}_j\|}$$

and corresponding absolute difficulty differences:

$$\Delta_{ij} = |y_i - y_j|.$$

We then group instance pairs into K bins based on quantiles of Δ_{ij} and compute the mean embedding distance within each bin. This results in a function $f(k)$ that maps increasing difficulty differences to average embedding distances. A monotonically increasing trend in $f(k)$ indicates that the representation space is structured according to difficulty.

F Cross-lingual Alignment Analysis Details

To evaluate whether the learned representations capture cross-lingual semantic alignment, we measure the cosine similarity between representations of semantically corresponding inputs across languages. For each instance x_i , we construct two views: an L1-based input and its corresponding English-based input. We then obtain normalized embeddings $\mathbf{h}_i^{L1}$ and $\mathbf{h}_i^{EN}$ using the encoder.

We define the similarity of aligned pairs as:

$$s_i^{\text{aligned}} = \mathbf{h}_i^{L1} \cdot \mathbf{h}_i^{EN},$$

and construct a baseline by randomly permuting the English representations to obtain mismatched pairs:

$$s_i^{\text{random}} = \mathbf{h}_i^{L1} \cdot \mathbf{h}_{\pi(i)}^{EN},$$

where $\pi(i)$ is a random permutation. By comparing the distributions of s^{aligned} and s^{random} , we assess whether the model places semantically corresponding inputs closer in the embedding space than unrelated ones.

Data Asgardians at BEA 2026 Shared Task 1: A Hybrid Transformer–Feature Ensemble for L1-Aware English Vocabulary Difficulty Prediction

Adrián Pineda Sánchez¹, Sabur Butt², Héctor Gibrán Ceballos Cancino²

¹School of Engineering and Sciences, Tecnológico de Monterrey, Mexico

²Institute for the Future of Education (IFE), Tecnológico de Monterrey, Mexico
A00834710@tec.mx, saburb@tec.mx, ceballos@tec.mx

Correspondence: Sabur Butt, saburb@tec.mx

Abstract

This paper presents our system for the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners (Felice and Skidmore, 2026). The task requires predicting psychometrically calibrated GLMM difficulty scores for English vocabulary items across three learner first-language (L1) backgrounds: Spanish (ES), German (DE), and Mandarin Chinese (CN). Our approach studies how hand-crafted linguistic features can complement contextual multilingual transformer representations. We engineer 33 phonological, morphological, semantic, contextual, and cross-lingual features, and evaluate feature-only regressors, Solo transformer models, Hybrid transformer models, and prediction-level ensembling. Our official Closed Track submissions were generated with XLM-RoBERTa-large Solo and Hybrid models, which improved over the official baseline for all three L1 groups, achieving test RMSEs of 1.182 (ES), 1.117 (DE), and 1.006 (CN) with a mean of 1.103. We then conducted a post-submission refinement using mDeBERTa-v3-base components and a Ridge stacking ensemble, which further reduced test RMSE to 1.037 (ES), 0.997 (DE), and 0.913 (CN), with a mean of 0.982, a mean improvement of 0.121 over our best XLM-RoBERTa-large system.

1 Introduction

Predicting the difficulty of English vocabulary items for second-language learners has direct applications in adaptive assessment, personalised content generation, and computer-adaptive testing. In educational measurement, item difficulty is commonly estimated with Generalised Linear Mixed Models (GLMM) (Baayen et al., 2008), which provide reliable psychometric calibration from learner response data. However, this process is costly to scale, since each new item bank typically requires extensive response collection. The

BEA 2026 Shared Task on Vocabulary Difficulty Prediction (Felice and Skidmore, 2026) addresses this limitation by framing difficulty estimation as a supervised NLP task over the British Council’s Knowledge-based Vocabulary Lists (KVL) (Skidmore et al., 2025), a multilingual resource that conditions vocabulary difficulty on the learner’s first language (L1).

Recent lexical complexity systems commonly fine-tune pre-trained language models directly on the regression objective (Shardlow et al., 2021). While effective, this approach may underuse linguistic factors known to shape vocabulary acquisition, such as frequency, phonological complexity, polysemy, morphological structure, and the distance between English and the learner’s L1 (Laufer and Goldstein, 2004). These factors suggest that L1-aware difficulty prediction should benefit from combining contextual transformer representations with explicit linguistic features.

Our system follows this feature-aware modelling perspective. We combine contextual multilingual representations with 33 engineered features covering phonological, morphological, semantic, contextual, and cross-lingual properties. Our official Closed Track submissions were based on XLM-R Large Solo and Hybrid models. We then developed a refined post-submission system based on mDeBERTa-v3-base, feature-only regressors, and a cross-validated Ridge stacking ensemble.

This paper makes three contributions. First, we present an L1-aware system for vocabulary difficulty prediction that integrates multilingual transformer representations with linguistically motivated features. Second, we compare feature-only regressors, Solo transformers, Hybrid feature-augmented transformers, and prediction-level ensembling across Spanish, German, and Mandarin Chinese. Third, we show that the strongest performance among our systems is obtained through prediction-level stacking, indicating that comple-

mentary model signals are more effective than relying on a single transformer or on direct feature concatenation alone.

2 Task and Data

The task data come from the Extended KVL Dataset (Skidmore et al., 2025), which provides GLMM-calibrated difficulty scores for English vocabulary items across three L1 backgrounds: Spanish (ES), German (DE), and Mandarin Chinese (CN). Each instance includes the English target word, its part of speech, a letter-based clue, an L1 translation, and an L1 context sentence. GLMM scores are continuous logit values, with higher values indicating easier items.

Table 1 shows representative examples across the three L1 groups, including the clue format, translation, context, and difficulty score.

As shown in Table 2, the dataset is balanced across L1 groups, with 6,091 training instances, 677 development instances, and 748 test instances per language. The Closed track restricts systems to the provided data and standard NLP tools, while the Open track allows external resources. Systems are ranked by RMSE, with Pearson’s r used as a secondary metric.

3 Related Work

The BEA 2026 Shared Task on vocabulary difficulty prediction (Felice and Skidmore, 2026) frames item difficulty estimation as a supervised regression problem over psychometrically calibrated GLMM scores. Unlike general lexical complexity benchmarks, the task requires predicting difficulty across multiple learner first-language (L1) groups, making the problem dependent not only on the English target word and its context, but also on cross-linguistic effects between English and the learner’s L1.

Recent lexical complexity systems have largely relied on pre-trained transformer models. In the SemEval-2021 shared task, Shardlow et al. (2021) showed that fine-tuned contextual models are strong predictors of lexical complexity, while traditional lexical features, especially frequency-based ones, can still provide complementary signal.

For L1-aware vocabulary difficulty, however, linguistic and psycholinguistic factors remain important. Prior work shows that word length, phonological similarity, morphological complexity, and related form-based properties affect vocabulary learn-

ing beyond frequency alone (Laufer and Goldstein, 2004). This suggests that difficulty is not only a property of the target word, but also of its relationship to the learner’s linguistic background.

Our work follows this perspective by combining contextual transformer representations with explicit phonological, morphological, semantic, and cross-lingual features. This hybrid setting allows the model to capture both usage-dependent difficulty and L1-specific signals under the limited-data conditions of the shared task.

4 Exploratory Data Analysis

Before modelling, we examine the GLMM target distribution and its structure across L1 groups under the Closed Track setting. This analysis informs two design choices: using linguistically motivated features, and modelling each L1 separately while still exploiting shared difficulty patterns.

Fig. 1 shows that GLMM scores are approximately symmetric and centred near zero for Spanish, German, and Mandarin Chinese, suggesting no major distributional imbalance across L1 groups. The distributions are similar overall, but differ slightly in dispersion (CN $\sigma \approx 1.26$, ES $\sigma \approx 1.43$, DE $\sigma \approx 1.38$), with Mandarin more concentrated and Mandarin/German showing more pronounced negative outliers.

We also find strong correlations for identical items across L1 groups (ES–DE: $r \approx 0.85$; ES–CN: $r \approx 0.78$; DE–CN: $r \approx 0.80$), indicating a shared global difficulty component. However, the remaining L1-specific variation motivates separate per-L1 models and feature engineering that captures cross-linguistic effects. Together, these observations support the hybrid modelling strategy developed in the following sections.

5 Feature Engineering

Motivated by the EDA findings, we design 33 linguistically informed features to capture both shared and L1-specific sources of vocabulary difficulty. The features are derived from WordNet (Miller, 1995), the CMU Pronouncing Dictionary (Carnegie Mellon University, 2014), and PHOIBLE (Moran and McCloy, 2019). Table 3 summarizes the resulting feature groups.

The feature set is organized into four complementary groups:

- **Phonological features** capture English pronunciation complexity, including consonant

L1	Target	POS	Clue	L1 translation	Context	GLMM
ES	<i>span</i>	noun	s__	lapso	El eclipse solar fue visible durante un breve ...	-3.264099
ES	<i>radically</i>	adverb	r_____	fundamentalmente	Los métodos nuevos son fundamentalmente difere...	-1.871782
DE	<i>span</i>	noun	s__	Zeitraum	Über einen Zeitraum von fünf Monaten wurden Ze...	-2.769408
DE	<i>radically</i>	adverb	r_____	fundamental, völlig	Nach unserem Gespräch hat sich seine Einstellu...	-1.567126
CN	<i>span</i>	noun	s__	持续时间段	智能手机使人的注意力集中时间越来越短。	-2.741183
CN	<i>radically</i>	adverb	r_____	彻底地，激进地，根本本地	我们应当彻底改变只顾经济发展的思路，以保护日渐脆弱的环境。	-1.337120

Table 1: Representative Extended KVL items across Spanish, German, and Mandarin Chinese. Higher GLMM values indicate easier items.

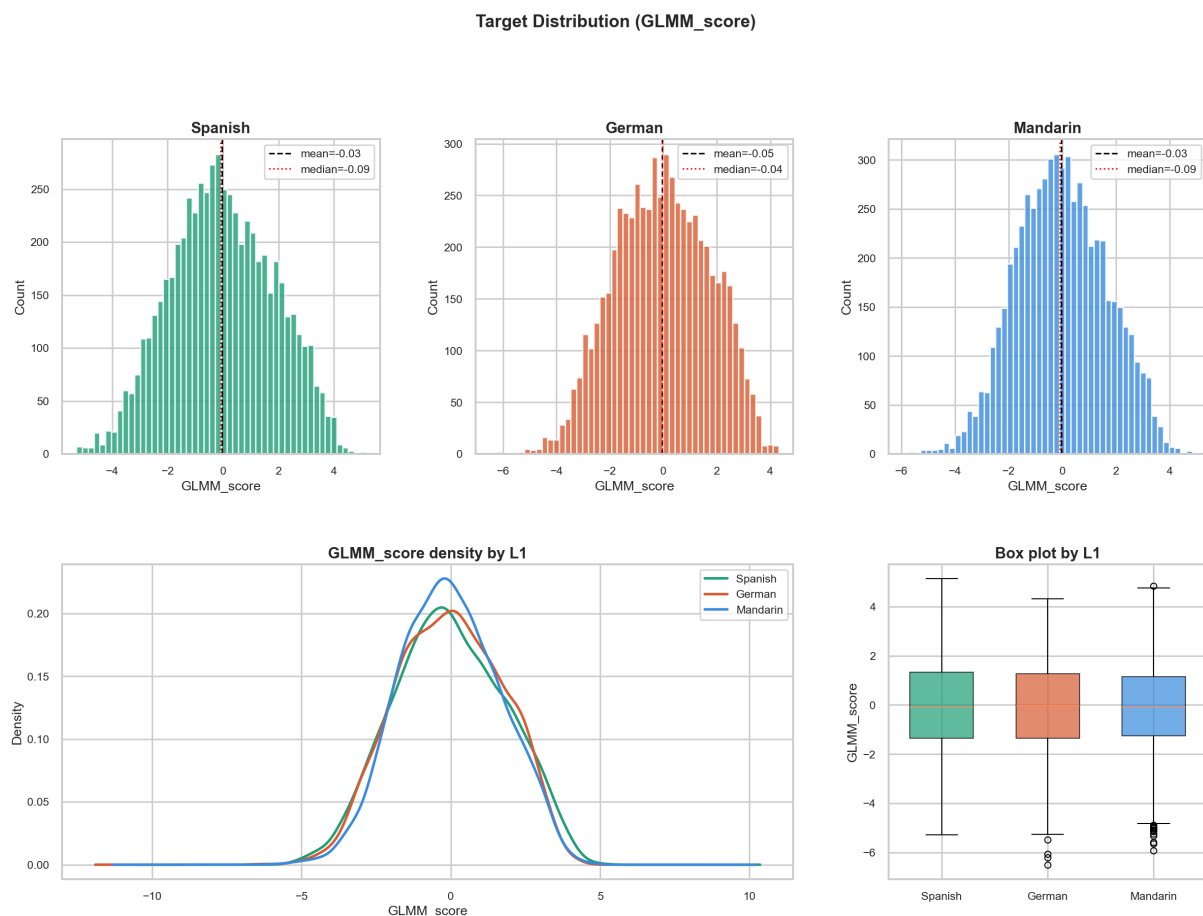


Figure 1: GLMM difficulty score distributions by L1 (training set).

L1	Train	Dev	Test
Spanish (ES)	6,091	677	748
German (DE)	6,091	677	748
Mandarin (CN)	6,091	677	748
Total	18,273	2,031	2,244

Table 2: Extended KVL dataset split sizes per L1.

clusters, word-final clusters, rhoticity, silent letters, and spelling-to-phoneme mismatch. The feature *r_count* counts /r/ phonemes in the CMU pronunciation, not occurrences of

the orthographic character *r*.

- **L1-specific phonological features** encode whether the English target contains sounds or graphemic patterns that may be absent or difficult for a given learner L1. For example, *difficult_phonemes* counts English phonemes not present in the learner’s L1 inventory according to PHOIBLE.
- **Semantic and morphological features** represent lexical properties from WordNet, including polysemy, homonymy, hypernym depth,

Category	#	Features	Source
Phonological	7	n_consonant_clusters, max_cluster_length, final_cluster_size, r_count, has_r, silent_letters, spelling_phoneme_ratio	CMU Dict
L1-specific phonological	5	difficult_phonemes, has_unfamiliar, has_diphthong, primary_stress_pos, has_l1_flag	PHOIBLE
Semantic & morphological	8	polysemy_pos, polysemy_all, is_homonym, homonym_pos_count, min_sense_depth, morphological_complexity, word_family_size, word_frequency	WordNet
Cross-lingual & contextual	13	cosine_dist_l1_en, clue_ratio, source_word_length, source_word_count, context_char_length, shared_prefix_len, pos_adjective, pos_adverb, pos_determiner, pos_misc, pos_noun, pos_number, pos_preposition, pos_verb	Text / WordNet
Total	33	25 numerical + 8 POS one-hot	–

Table 3: Engineered feature groups and sources. `cosine_dist_l1_en` is a character bigram Jaccard distance between the L1 form and the English target.

morphological complexity, word family size, and lemma frequency. These features approximate how semantically broad, frequent, or morphologically complex a target word is.

- **Cross-lingual and contextual features** describe the relation between the English target and the L1-side information. This includes `cosine_dist_l1_en`, implemented as character bigram Jaccard distance between the L1 translation and the English target, as well as clue ratio, source-word length, context length, shared prefixes, and POS indicators. The feature `clue_ratio` is defined as the proportion of orthographic characters of the English target that are revealed by the letter-based clue, i.e., the ratio between the number of revealed characters (non-underscore positions) and the total length of the target word. Higher values correspond to more informative clues and, empirically, to easier items (see Table 4).

To reduce redundancy, we compute pairwise Pearson correlations among candidate features on the training set. For any pair with $|r| > 0.85$, we remove the feature with the lower absolute correlation with the GLMM target. This process yields the final 33-feature set with no remaining pair above the collinearity threshold.

Table 4 reports the strongest feature–target correlations. `clue_ratio` and `word_frequency` are the most predictive global features, while the per-L1 values show that feature relevance varies across learner backgrounds. Figure 2 extends this analysis to all 33 features, highlighting both globally predictive signals and L1-specific variation.

6 System Architecture

Our official Closed-track prediction files were generated with XLM-RoBERTa-large Solo and Hybrid

Feature	Global	ES	DE	CN
<code>clue_ratio</code>	−0.381	−0.350	−0.353	−0.449
<code>word_frequency</code>	0.314	0.237	0.233	0.488
<code>source_word_length</code>	−0.253	−0.217	−0.311	−0.338
<code>primary_stress_pos</code>	−0.247	−0.216	−0.258	−0.271
<code>cosine_dist_l1_en</code>	−0.246	−0.271	−0.352	0.060

Table 4: Top-5 features by absolute Pearson r with the GLMM difficulty score, computed globally and per L1 on the training set.

models. We then explored whether a smaller multilingual encoder, combined with structured linguistic features and prediction-level ensembling, could improve performance. The refined system is built around mDeBERTa-v3-base and combines four learners through a cross-validated Ridge stacking ensemble (Wolpert, 1992): (i) a Solo transformer, (ii) a Hybrid transformer, (iii) an Optuna-tuned XGBoost regressor, and (iv) a Ridge regressor. We treat this ensemble as the primary system and report XLM-R Large as the official-submission reference.

6.1 Training Protocol

All systems follow the official train/development/test splits (Table 2) and are trained independently for each L1 group (ES, DE, CN), following the L1-specific variation observed in Section 4. Model development follows two phases. First, models are trained on the official training set and selected on the development set using RMSE, with Pearson’s r as a secondary metric. This phase is also used for hyperparameter search and, in the stacking ensemble, for selecting the Ridge meta-learner’s regularisation strength α using 5-fold out-of-fold RMSE. Second, the selected configuration is retrained on train+dev

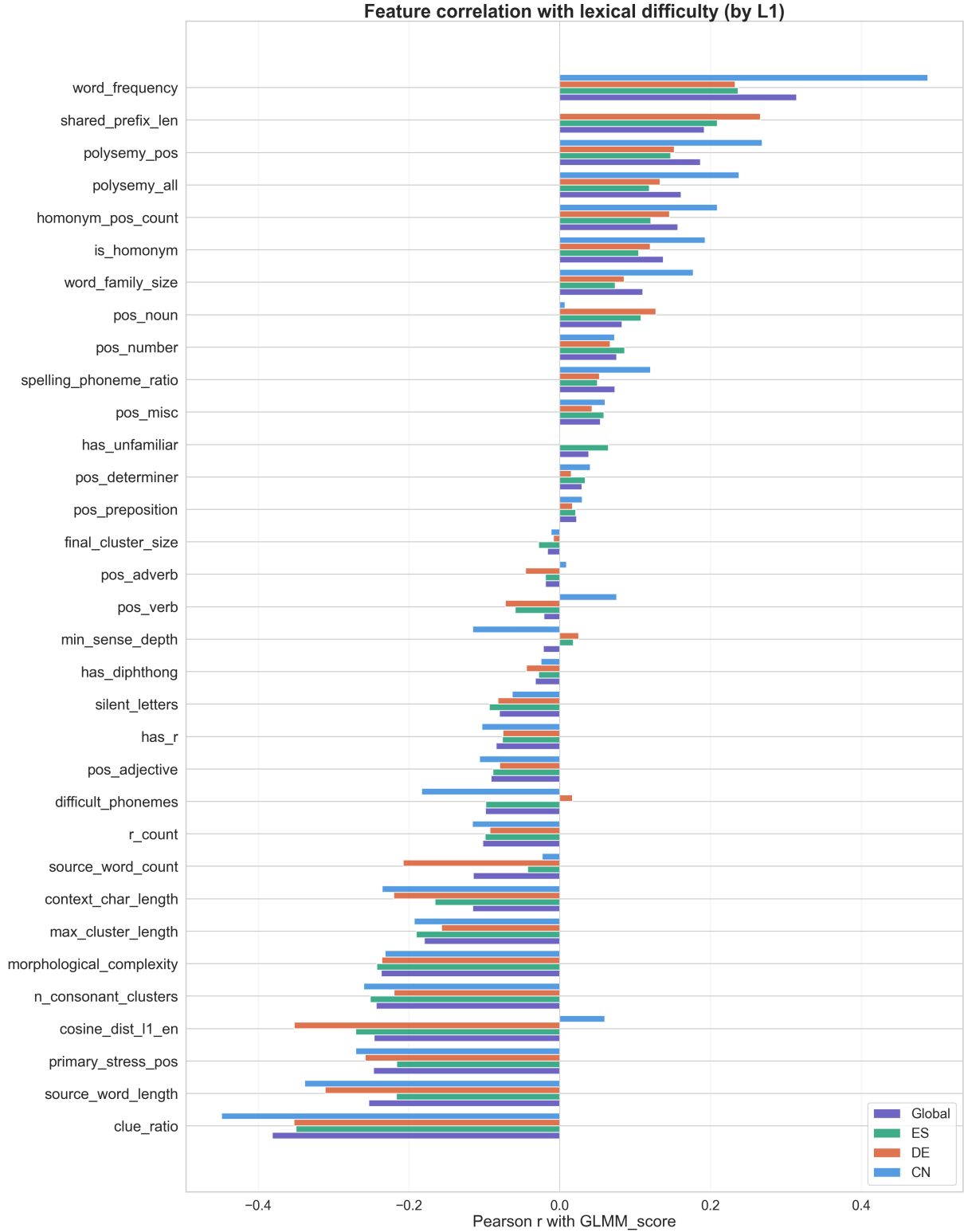


Figure 2: Pearson r between each engineered feature and the GLMM score (training set), computed globally and per L1.

and applied to the test set.

For the Solo and Hybrid transformer variants, we use the same per-L1 grid: learning rate $\{5 \times 10^{-6}, 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$, weight decay $\{0.01, 0.05, 0.10\}$, dropout $\{0.10, 0.15\}$,

and epochs $\{3, 5, 7, 8, 10\}$. This produced 42 mDeBERTa-v3-base runs and 33 XLM-R Large runs. The selected configurations are shown in Table 5. Feature-only XGBoost and Ridge models were tuned with Optuna as described in Sec-

tion 6.4. All transformer experiments were run on rented Google Colab GPU compute, mainly NVIDIA A100 40GB when available, requiring approximately 200 GPU hours.

Encoder	L1	Model	lr	wd	drop	ep
mDeBERTa-base	ES	Solo	3×10^{-5}	0.05	0.10	5
	ES	Hybrid	1×10^{-5}	0.10	0.15	5
	DE	Solo	1×10^{-5}	0.05	0.10	7
	DE	Hybrid	1×10^{-5}	0.10	0.15	5
	CN	Solo	1×10^{-5}	0.10	0.15	10
	CN	Hybrid	2×10^{-5}	0.05	0.10	3
XLM-R Large	ES	Solo	2×10^{-5}	0.01	0.10	8
	ES	Hybrid	5×10^{-6}	0.10	0.15	5
	DE	Solo	1×10^{-5}	0.05	0.10	10
	DE	Hybrid	1×10^{-5}	0.01	0.10	5
	CN	Solo	2×10^{-5}	0.01	0.10	8
	CN	Hybrid	1×10^{-5}	0.05	0.10	5

Table 5: Best hyperparameter configuration per encoder, L1, and model type (Solo vs. Hybrid), selected on the development set. lr: learning rate; wd: weight decay; drop: dropout rate; ep: training epochs.

6.2 Solo Transformer

We evaluate two multilingual encoders: XLM-RoBERTa-large (Conneau et al., 2020), used in the official submission, and mDeBERTa-v3-base (He et al., 2023), used in the refined system. Both models are used in their publicly released pre-trained form, without intermediate task-adaptive pre-training, and are fine-tuned under the same protocol. Hyperparameters are selected independently per L1 (Table 5).

Each encoder is paired with a regression head over the first-token representation of the last hidden layer:

$$\mathbf{h} = \mathbf{H}_{[0]}, \quad \hat{y} = \mathbf{W} \text{Dropout}(\mathbf{h}), \quad (1)$$

where $\mathbf{H} \in \mathbb{R}^{|T| \times d}$ denotes the sequence of last-layer hidden states. The input concatenates the L1-SOURCE-WORD, L1-CONTEXT, EN-TARGET-CLUE, and EN-TARGET-WORD, tokenised with the encoder’s native SentencePiece tokeniser using a maximum length of 192. This format allows the model to jointly attend to the L1 context and the English target information.

6.3 Hybrid Transformer

The Hybrid model augments the transformer representation $\mathbf{h}$ with a 33-dimensional linguistic feature vector $\mathbf{f}$, using median imputation for missing

values. The concatenated representation is passed through a two-layer MLP:

$$\hat{y} = \mathbf{W}_2 \text{Dropout}(\phi(\mathbf{W}_1 \text{Dropout}([\mathbf{h}; \mathbf{f}]))) , \quad (2)$$

with $\mathbf{W}_1 \in \mathbb{R}^{256 \times (d+33)}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times 256}$. The hidden size is $d = 768$ for mDeBERTa-v3-base and $d = 1024$ for XLM-R Large. We use GELU for mDeBERTa-v3-base and ReLU for XLM-R Large.

Both encoders are fine-tuned with AdamW, mixed-precision training (fp16), a linear-warmup and linear-decay schedule with a 10% warmup ratio, gradient clipping at 1.0, and an MSE objective. Learning rate, weight decay, dropout, and number of epochs are tuned per L1.

6.4 Feature-Only Regressors

We also train XGBoost (Chen and Guestrin, 2016) and Ridge regression models directly on the 33 engineered linguistic features. Both are trained independently per L1, using median imputation; Ridge additionally uses standardisation. Hyperparameters are selected with Optuna (Akiba et al., 2019), using 150 trials per L1 for XGBoost and 100 trials per L1 for Ridge. These models serve both as ensemble components and as feature-only baselines.

6.5 Ridge Stacking Ensemble

The refined system combines the four mDeBERTa-v3-base components: Solo ($\hat{y}_S$), Hybrid ($\hat{y}_H$), XGBoost ($\hat{y}_X$), and Ridge ($\hat{y}_R$). For each L1, a Ridge meta-learner combines their predictions:

$$\hat{y}_{\text{ens}} = \mathbf{w}^\top [\hat{y}_S, \hat{y}_H, \hat{y}_X, \hat{y}_R], \quad (3)$$

with non-negative weights ($w_i \geq 0$). The regularisation strength α is selected from 40 logarithmically spaced values in $[10^{-2}, 10^3]$ using 5-fold out-of-fold RMSE on the development set. The meta-learner is then refit on the full development set and applied to the test set. The XLM-R Large reference systems are not included in this ensemble and are reported separately. The full training protocol, including the two-phase train+dev re-training scheme and the grid search ranges shared with the Solo and Hybrid models, is described in Section 6.1.

7 Results

Table 6 reports our official Closed Track submissions. All submitted runs used XLM-R Large and

Lang.	Run	Pos.	RMSE	Pear.	Δ RMSE
<i>Spanish</i> – Baseline RMSE = 1.257 $r = 0.765$					
ES	Solo (tuned)	33/59	1.182	0.820	+0.075
ES	Hybrid (tuned)	34/59	1.186	0.825	+0.071
ES	Solo (base)	35/59	1.190	0.814	+0.067
<i>German</i> – Baseline RMSE = 1.258 $r = 0.773$					
DE	Hybrid (tuned)	26/56	1.117	0.830	+0.141
DE	Solo (base)	35/56	1.140	0.829	+0.118
DE	Solo (tuned)	40/56	1.177	0.822	+0.081
<i>Chinese</i> – Baseline RMSE = 1.140 $r = 0.753$					
CN	Hybrid (tuned)	26/51	1.006	0.854	+0.134
CN	Solo (tuned)	28/51	1.008	0.850	+0.132
CN	Solo (base)	29/51	1.013	0.842	+0.127

Table 6: Official Closed Track test results. Positive Δ RMSE indicates improvement over the baseline; bold marks the best value per L1 group.

improved over the official baseline for the three L1 groups. The largest gains were obtained for German and Chinese, where the Hybrid model achieved the best official RMSE. For Spanish, the Solo model was slightly stronger.

Table 7 compares the official systems with the post-submission refinements and includes development-set RMSE. The XLM-R Large Hybrid model was the strongest single model on development, but the mDeBERTa stacking ensemble achieved the lowest RMSE on both development and test. On the test set, the ensemble obtained 1.037 for Spanish, 0.997 for German, and 0.913 for Chinese, reducing the mean RMSE from 1.103 for the best official XLM-R Large system to 0.982.

Feature-only models did not outperform the baseline, despite the correlations reported in Table 4. This suggests that features such as `clue_ratio`, `word_frequency`, `source_word_length`, and `primary_stress_pos` capture useful but incomplete signal. Transformer models provided stronger contextual representations, while prediction-level stacking benefited from combining transformer and feature-based signals.

Direct feature fusion helped XLM-R Large but not mDeBERTa. The XLM-R Large Hybrid improved over Solo in mean test RMSE (1.103 vs. 1.122), whereas the mDeBERTa Hybrid was weaker than mDeBERTa Solo (1.150 vs. 1.110). One possible explanation is that the 33-dimensional feature vector is a relatively small addition to the 1024-dimensional XLM-R Large representation, but a more influential input to the smaller mDeBERTa head. The stacking ensemble avoids this issue

by combining feature-based and transformer-based predictions only at the output level.

Figure 3 visualizes the same pattern across Spanish, German, and Mandarin Chinese. The final mDeBERTa Ensemble is consistently the lowest-error system, supporting the conclusion that prediction-level ensembling is more robust than relying only on direct feature fusion or a single model family.

8 Conclusions

We presented a system for the BEA 2026 Shared Task on vocabulary difficulty prediction, structured around an official Closed Track submission and a post-submission refinement. The official submissions used XLM-R Large Solo and Hybrid models and improved over the baseline for Spanish, German, and Mandarin Chinese. A post-submission refinement based on mDeBERTa-v3-base, feature-only regressors, and Ridge stacking further reduced error, achieving the best RMSE among our systems for all L1s: 1.037 (ES), 0.997 (DE), and 0.913 (CN), with a mean RMSE of 0.982.

The results show that engineered features contributed useful linguistic signal, although they were most effective when combined with transformer-based predictions rather than used as standalone models or through direct Hybrid concatenation. Although the final ensemble was not part of the official submission, its estimated ranks suggest a substantial improvement over our official systems, reaching approximately 8/60 for Spanish, 11–12/57 for German, and 15/52 for Chinese. Overall, L1-aware vocabulary difficulty prediction benefits most from combining complementary model signals at the prediction level rather than relying on a single model family or feature fusion strategy alone.

Limitations

This study is limited to the Extended KVL dataset and three L1 backgrounds (Spanish, German, and Mandarin Chinese), so generalization to other learner groups and vocabulary benchmarks remains future work. The hyperparameter search was constrained by available GPU time, and the ensemble was not evaluated under fully nested cross-validation. Finally, our analysis focuses mainly on RMSE and Pearson’s r ; future work should include finer-grained error analysis by word type, POS, clue structure, and L1-specific properties.

System	Submission	Dev RMSE			Test RMSE			Mean	Rank (ES/DE/CN)
		ES	DE	CN	ES	DE	CN		
XLM-R Base Baseline	—	1.357	1.328	1.175	1.257	1.258	1.140	1.218	—
XGBoost (Optuna)	Post-sub.	1.471	1.427	1.285	1.461	1.351	1.279	1.364	—
Ridge (Optuna)	Post-sub.	1.540	1.477	1.354	1.505	1.407	1.293	1.402	—
XLM-R Large Solo	Official	1.139	1.126	1.022	1.182	1.177	1.008	1.122	33/40/28
XLM-R Large Hybrid	Official	1.118	1.101	1.019	1.186	1.117	1.006	1.103	34/26/26
mDeBERTa Solo	Post-sub.	1.153	1.172	1.057	1.152	1.141	1.037	1.110	—
mDeBERTa Hybrid	Post-sub.	1.181	1.173	1.054	1.180	1.157	1.112	1.150	—
mDeBERTa Ensemble	Post-sub.	1.068	1.065	0.981	1.037	0.997	0.913	0.982	≈8/11–12/15

Table 7: Development and test RMSE for all evaluated systems. Bold marks the best score in each column; ensemble ranks are estimated, not official.

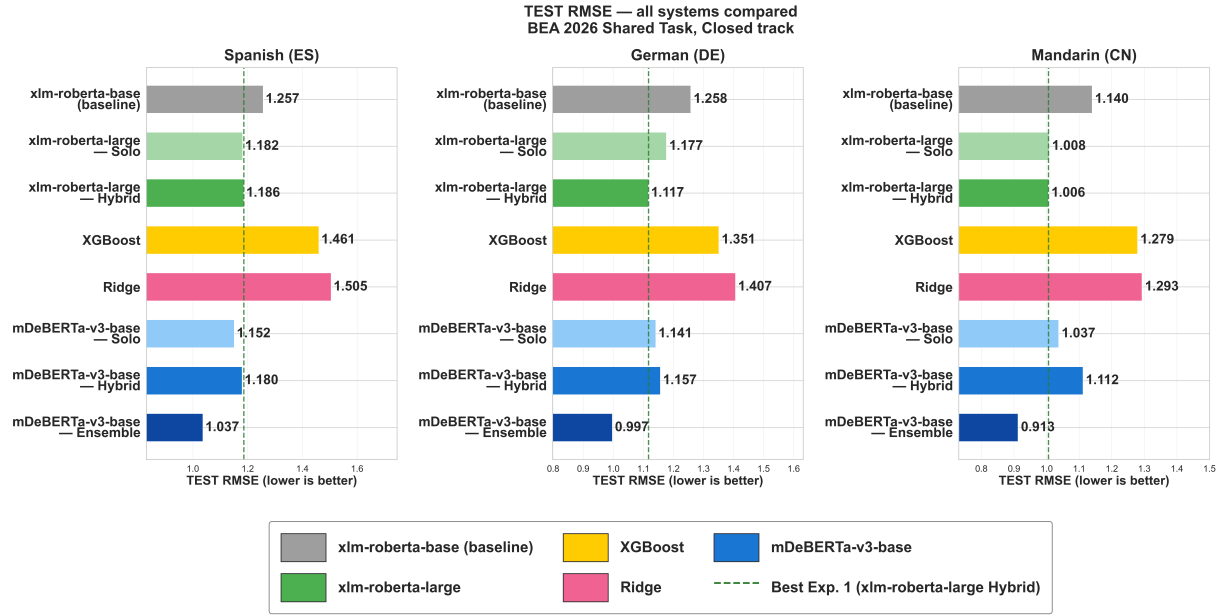


Figure 3: Test RMSE per system and L1 group. Lower is better; the dashed line marks the official XLM-R Large Hybrid reference.

A Code and Reproducibility

Code, notebooks, feature pipelines, model configurations, figures, prediction files, and output tables are available at: https://github.com/AdrianPinedaSanchez/BEA2026_shared_task_DataAsgardians

References

- Takuya Akiba, Shotaro Sano, Toshihiro Yanase, Takeru Ohta, and Masanori Koyama. 2019. *Optuna: A next-generation hyperparameter optimization framework*. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631. ACM.
- R. Harald Baayen, Douglas J. Davidson, and Douglas M. Bates. 2008. *Mixed-effects modeling with crossed random effects for subjects and items*. *Journal of Memory and Language*, 59(4):390–412.
- Carnegie Mellon University. 2014. *The CMU pronouncing dictionary, v0.7b*.
- 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. ACM.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. *Unsupervised cross-lingual representation learning at scale*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.
- Mariano Felice and Lucy Skidmore. 2026. Findings of

- the BEA 2026 shared task on vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. 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](#). In *Proceedings of the 11th International Conference on Learning Representations*.
- Batia Laufer and Zahava Goldstein. 2004. [Testing vocabulary knowledge: Size, strength, and computer adaptiveness](#). *Language Learning*, 54(3):469–523.
- George A. Miller. 1995. [WordNet: A lexical database for English](#). *Communications of the ACM*, 38(11):39–41.
- Steven Moran and Daniel McCloy. 2019. [PHOIBLE 2.0](#).
- 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. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- David H. Wolpert. 1992. [Stacked generalization](#). *Neural Networks*, 5(2):241–259.

UOL@IDEM at BEA 2026 Shared Task 1: Neural Fusion and Feature-Rich Modeling for L1-Aware Vocabulary Difficulty Prediction

Nouran Khallaf^{1,2} Serge Sharoff¹

¹Centre for Translation, Localisation and Interpreting Studies,
School of Languages, Cultures and Societies, University of Leeds, UK

²Alexandria University, Egypt
{n.khallaf,s.sharoff}@leeds.ac.uk

Abstract

This paper describes UOL@IDEM’s closed-track submission to the BEA 2026 shared task on L1-aware vocabulary difficulty prediction¹. We model the task as regression and train separate systems for Spanish, German, and Mandarin Chinese². Our system combines multilingual contextual representations with engineered features capturing frequency, surface form, retrieval evidence, semantic alignment, cognate similarity, and masked-language-model predictability. Development results show consistent gains over the official closed-track baselines, with sentence-embedding encoders such as BGE-M3, multilingual E5, and LaBSE performing best. Official submissions achieve RMSE scores of 1.132, 1.037, and 0.891 for Spanish, German, and Chinese, respectively. Feature analysis identifies frequency as the most stable predictor, while contextual predictability, form similarity, retrieval, and semantic features provide complementary L1-sensitive signals. Error analysis shows strong ranking performance but weaker calibration for the easiest items, which are often overpredicted.

1 Introduction

Text complexity and readability assessment are central to educational NLP, language learning, and text adaptation. A key component is lexical complexity prediction (LCP), which estimates how difficult a word or multi-word expression is likely to be for a target reader (Shardlow et al., 2021, 2022; North et al., 2023). Lexical difficulty is shaped by frequency, word length, morphology, polysemy, context, and reader background, and supports applications such as readability assessment, simplification, tutoring, and machine translation (Shardlow et al., 2022; North et al., 2023; Oluoba et al., 2024).

¹<https://github.com/Nouran-Khallaf/>

UoL-IDEM-BEA2026-Vocabulary-Difficulty-Prediction

²Below we use *Chinese* for brevity.

The BEA 2026 Shared Task extends LCP by making vocabulary difficulty explicitly L1-aware. In this setting, the difficulty of an English target word depends not only on the word itself, but also on the context of its use as well as on its relationship to the learner’s first language through translation and transfer cues. As a result, the same English word may have different difficulty profiles for Spanish, German, and Chinese speakers. More specifically, the task uses an extended version of the Knowledge-based Vocabulary Lists (KVL) resource, pairing English target items with multilingual prompts and psychometrically calibrated GLMM-based difficulty scores (Skidmore et al., 2025).

In the closed track, systems train separate models for each L1 using only the shared-task resources (Felice and Skidmore, 2026; British Council, 2026). We therefore model the task as three parallel prediction settings: Spanish→English, German→English, and Chinese→English.

Our approach experiments with mid-sized text encoders (of up to 560M parameters) enriched with engineered features, such as lexical frequency, orthographic clues, contextual retrieval, cognate-like similarity, semantic-domain alignment, and masked-language-model predictability. We evaluate several encoders within a neural fusion framework and compare full-feature and reduced-feature variants.

Our experiments show three main findings. First, neural fusion improves over the official closed-track baselines across all three languages, improving RMSE by up to 0.21–0.26. Second, sentence-embedding-oriented multilingual models, especially BGE-M3 (Chen et al., 2024), multilingual E5 (Wang et al., 2024), and LaBSE (Feng et al., 2022), are more effective than plain transformer baselines such as mBERT and XLM-R. Third, feature and error analyses show that frequency is the most stable predictor contributing above the en-

coder baseline, while semantic, cognate, retrieval, and MLM-based cues can provide complementary L1-sensitive information.

2 Data and Preprocessing

Data exploration showed that L1_source_word often contains noisy forms that require language-aware normalisation before feature extraction. We address lexical variants, negative constraints, morphological fragments, negated instructional patterns, explanatory glosses, and punctuation. For example, multiple alternatives are split and tracked with a variant-count feature (*Korridor, Flur* → *Korridor*); exclusion notes are removed and encoded as binary features (*Flugzeug (nicht: ...)* → *Flugzeug*); truncated German compounds are normalised to stems (*Regierungs-* → *Regierungs*); Spanish instruction-like phrases such as *no es* and *la respuesta no termina en* are stripped to isolate the lexical item; and Chinese punctuation, leading glosses, and parenthetical notes are pruned where appropriate ((美) 走廊, 过道 → 走廊). To make preprocessing traceable, we keep the original source-word string and use a cleaned lexical form for feature extraction. When an English target contains multiple candidates, we select the candidate that appears in the L1 context.

3 Methodology

We formulated the task as supervised regression. Each instance contained an L1-specific context, an L1 source word, an English target word, and a continuous difficulty score. Models used textual input and engineered features capturing complementary cues: frequency, lexical form, retrieval evidence, MLM and surprisal estimates, semantic-domain information, and cognate-based similarity.

We compared three integration strategies: (1) textualised feature regression, which renders selected features as part of the input prompt; (2) multi-stage late fusion, which combines frozen encoder embeddings with downstream regressors; and (3) neural fusion, which jointly optimises contextual representations and structured tabular features. Because neural fusion performed best in the initial comparison, we used it as the main setting.

3.1 Engineered features

In addition to BERT-derived text representations, we designed a set of engineered features to capture lexical difficulty, cross-lingual retrievability,

contextual predictability, semantic compatibility, and source-word ambiguity. We group these features into frequency-based, lexical, retrieval-based, masked-language-model and surprisal, semantic, and cognate-based features. Full feature descriptions are provided in Appendix A, Table 5.

Frequency-based features. We combine three complementary frequency resources. KELLY (Kilgarriff et al., 2014) is derived from curated Web-corpora and provides pedagogically oriented indicators, including frequency rank, percentile, and CEFR-aligned labels, which approximate learner vocabulary progression (Council of Europe, 2001). The wordfreq Python package provides general corpus-based estimates, including raw frequency, Zipf frequency, frequency percentile, and inverse-frequency cost (Speer et al., 2018; Sharoff et al., 2017). The SUBTLEX list is derived from subtitles and provides frequency and contextual-diversity measures, reflecting both occurrence frequency and exposure across everyday contexts (Brysbaert and New, 2009; van Heuven et al., 2014). Combined, these features model familiarity from pedagogical, general-corpus, and everyday-exposure perspectives, providing a strong baseline signal for vocabulary difficulty prediction (Shardlow et al., 2021, 2022; North et al., 2023).

Surface-form and morphosyntactic features.

We extract surface-form and morphosyntactic features from the English target and cleaned L1 source word. These include target/source length, target syllable count, clue informativeness, and hidden clue characters. We also encode POS information, using the dataset-provided target POS and spaCy-derived source POS (Honnibal et al., 2020). Alongside these features, we also retain a set of preprocessing-derived indicators introduced in Section 2, including signals for exclusion notes, alternative candidate forms, and related counts.

Retrieval-based features. We derive retrieval-based features by treating the task as context-guided cross-lingual retrieval of the English target. We build a fixed English candidate bank from possible target words, encode candidates with mBERT (Devlin et al., 2019), and map them into a shared multilingual embedding space. For each instance, the L1 context and source word are encoded as a source-side representation and matched against this bank. Candidates are reranked using the available constraints: initial-letter clue, target length, and

part of speech. Matching candidates are promoted, while mismatches are downweighted or excluded. We then extract features describing gold-target support, including rank before and after reranking, cosine and reranked scores, top-candidate scores, margins, nearest-neighbour confidence and entropy, gold-target probability measures, and the number of valid candidates. The gold target’s retrieval position is encoded as its rank when retrieved and as 0 otherwise. These features capture how recoverable the target is from the L1 context and source word.

Masked language model and surprisal features.

We use the MLM head of mBERT (Devlin et al., 2019) to estimate how predictable the English target is in context. Unlike retrieval features, which compare the target with an external candidate bank, MLM features come from the output distribution at the masked target position. We apply target constraints, especially initial letter and length, so the scores reflect both contextual fit and target-form compatibility. Features include gold-target log probability and rank, masked-token entropy, top-1 indicators, prediction margins, target–prediction embedding similarity, and related confidence measures.

Because targets may contain multiple word-pieces, probability-based quantities are computed at the subword level and averaged across the target span. For a target sequence $t_{1:m}$, we use three surprisal variants. Simultaneous masking replaces the full target span with [MASK]:

$$s_{\text{masked}}(t_{1:m}) = -\frac{1}{m} \sum_{j=1}^m \log P(t_j \mid \mathbf{c}_{\text{mask}}). \quad (1)$$

Pseudo-log-likelihood (PLL) masks one target wordpiece at a time while keeping the remaining pieces visible (Salazar et al., 2020):

$$s_{\text{PLL}}(t_{1:m}) = -\frac{1}{m} \sum_{j=1}^m \log P(t_j \mid \mathbf{c}^{(j)}), \quad (2)$$

where $\mathbf{c}^{(j)}$ masks only token t_j , allowing each token to be predicted with access to the other target subwords.

Finally, chain-rule surprisal scores the target incrementally from left to right:

$$s_{\text{chain}}(t_{1:m}) = -\frac{1}{m} \sum_{j=1}^m \log P(t_j \mid \mathbf{c}, t_{1:j-1}). \quad (3)$$

These measures capture contextual predictability under full-span masking, single-token masking, and sequential prediction.

Semantic features. We derive semantic-domain features from the UCREL Semantic Analysis System (USAS), which assigns lexical items to 21 major discourse fields and 232 finer-grained categories (Rayson et al., 2004). For each item, we annotate the L1 source word and English target with all available USAS tags. We include overlap features for identical tags, shared fine-grained prefixes, and shared major fields. We also compute weighted and unweighted tag entropy to capture semantic ambiguity, with higher entropy indicating less stable interpretation.

To capture relatedness beyond exact overlap, we compute a semantic-shift score. Let S and T be the USAS tag sets for the L1 source and English target. For each source–target tag pair, we assign a graded relation score based on the strongest relation: exact tag overlap, shared fine-grained prefix, shared major field, related major fields, or unrelated fields. Item-level soft similarity is computed over the full tag sets, and semantic shift is defined as its complement:

$$\text{shift}_{\text{sem}}(S, T) = 1 - \text{sim}_{\text{soft}}(S, T). \quad (4)$$

Lower values indicate stronger semantic alignment; higher values indicate larger semantic-domain shift. For example, *obra* (‘literary work’ in Spanish) has the semantic tags of A1.1.1|I3.1, which do not overlap with *comedy* E4.1+|K4|Q4.3. Still, both refer to related fields, giving soft similarity 0.20 and semantic shift 0.80. Coverage gaps are handled using fixed penalties for unmatched Z99 cases. Appendix C, Table 7 gives further examples with full USAS category names.

Cognate features. We derive cognate-oriented features to capture lexical-transfer signals between the L1 source word and the English target. These include orthographic similarity, multilingual embedding cosine similarity, Levenshtein edit-distance similarity, character n -gram overlap, and prefix/suffix overlap. We also add lexical-relation features from CogNet, a multilingual cognate database (Batsuren et al., 2019, 2022). Because the task covers English, Spanish, German, and Chinese, we restrict CogNet to entries involving these languages. These features are expected to be most informative

for Spanish–English and German–English, while also testing whether cognate relations provide useful signals for Chinese–English.

3.2 Model Architectures

Textualised Feature Regression. The main text fields and selected engineered features are rendered as text, encoded with a multilingual transformer, and passed to a regression head. This allows feature–context interactions to be modelled through self-attention.

Multi-stage Late Fusion. Here the available text fields are concatenated and encoded with a frozen multilingual sentence encoder to obtain dense representations $\mathbf{h} \in \mathbb{R}^d$. These are used alone or concatenated with engineered features $\mathbf{x}$ as input to Ridge, GBDT, XGBoost, and SVR regressors. Their predictions are combined through ensemble-style aggregation (van der Laan et al., 2007), following feature-based regression approaches in quality estimation and related prediction tasks (Rios and Sharoff, 2016).

Neural Fusion. This architecture jointly learns from contextual text representations and structured tabular features. For each instance, the input sequence S_i concatenates L1_context, L1_source_word, en_target_word, and, in some configurations, an additional target clue using [SEP] tokens.

A multilingual transformer f_θ encodes S_i , and its token-level representations are pooled into a text vector:

$$\mathbf{h}_i = \text{Pool}(f_\theta(S_i)). \quad (5)$$

In parallel, a feed-forward tabular branch maps the engineered feature vector $\mathbf{x}_i$ into a dense representation $\mathbf{x}'_i$. The two representations are concatenated:

$$\mathbf{z}_i = [\mathbf{h}_i; \mathbf{x}'_i], \quad (6)$$

and passed to an MLP to predict the difficulty score:

$$\hat{y}_i = \text{MLP}(\mathbf{z}_i). \quad (7)$$

Text representations vary by model family: neural fusion uses attention pooling over final hidden states, textualised regression uses the standard [CLS] representation, and late fusion uses dense sentence embeddings, optionally followed by PCA.

3.3 Experimental Setup and Optimisation

We train separate models for Spanish, German, and Chinese. For each language, we evaluate multilingual PLMs, including XLM-R and mBERT (Conneau et al., 2020; Devlin et al., 2019), and sentence-oriented multilingual encoders, including LaBSE, Multilingual-E5, and BGE-M3 (Feng et al., 2022; Wang et al., 2024; Chen et al., 2024).

All experiments use five-fold cross-validation, with out-of-fold predictions used for training-set evaluation and held-out embeddings used for late fusion to avoid leakage. Models are optimised with AdamW and early stopping on validation RMSE, with differential learning rates, warmup, gradient clipping, and mixed precision. We test both mean squared error and Huber loss, the latter reducing the influence of large residuals; details are given in Appendix Eq. 8 and Appendix B, Table 6. RMSE is the primary metric used by the shared task organisers, while Pearson, Spearman, and Kendall’s τ used to assess fit and ranking quality of each method.

4 Results

Table 1 reports the main development-set results for the closed track, comparing Textualised Feature Regression, Multi-stage Late Fusion, and Neural Fusion. Across all three languages, the best systems substantially outperform the official closed-track baselines, with RMSE gains of 0.2618 for Spanish using BGE-M3, 0.2407 for German using multilingual-E5-large, and 0.2069 for Chinese using BGE-M3.

The results show a clear advantage for sentence-embedding-oriented multilingual encoders. BGE-M3 and multilingual-E5-large, both based on XLM-RoBERTa-large, are consistently stronger, while LaBSE, a multilingual BERT-based sentence embedding model, remains competitive. This suggests that the main advantage comes not only from the underlying multilingual backbone, but also from training the model specifically for sentence-level cross-lingual representation. In contrast, the standard token encoders are weaker, mBERT is generally less competitive, and XLM-R is less stable, performing reasonably well for Spanish and German but falling below the baseline for Chinese.

For the Late Fusion and Textualised Feature Regression settings, we therefore used the two strongest sentence-embedding encoders, BGE-M3 and multilingual-E5-large, and report the best

Language	System	RMSE↓	Δ RMSE vs Closed ↑	Pearson ↑	Spearman ↑	Kendall ↑
es	Baseline (closed)	1.3570	—	0.7480	—	—
es	LaBSE	1.1732	0.1838	0.8116	0.8136	0.6189
es	XLM-R-large	1.1189	0.2381	0.8291	0.8340	0.6416
es	mBERT	1.3269	0.0301	0.7415	0.7459	0.5475
es	multilingual-E5	1.1134	0.2436	0.8301	0.8335	0.6386
es	BGE-M3	1.0952	0.2618	0.8324	0.8373	0.6473
es	Late fusion	1.2667	0.0903	0.7513	0.7583	0.5620
es	Textualised Feature	1.1261	0.2309	0.8214	0.8239	0.6324
de	Baseline (closed)	1.3280	—	0.7530	—	—
de	LaBSE	1.1568	0.1712	0.8015	0.8139	0.6208
de	XLM-R-base	1.2211	0.1069	0.7885	0.8058	0.6093
de	mBERT	1.2693	0.0587	0.7481	0.7640	0.5665
de	multilingual-E5-large	1.0873	0.2407	0.8234	0.8414	0.6446
de	BGE-M3	1.1457	0.1823	0.8135	0.8260	0.6284
de	Late fusion	1.2627	0.0653	0.7219	0.7365	0.5389
de	Textualised Feature	1.0961	0.2319	0.8252	0.8417	0.6466
cn	Baseline (closed)	1.1750	—	0.7360	—	—
cn	LaBSE	0.9937	0.1813	0.8179	0.8220	0.6343
cn	XLM-R-large	1.2552	-0.0802	0.6704	0.6699	0.4906
cn	mBERT	1.1227	0.0523	0.7673	0.7737	0.5868
cn	multilingual-E5	0.9726	0.2024	0.8340	0.8387	0.6493
cn	BGE-M3	0.9681	0.2069	0.8351	0.8428	0.6568
cn	Late fusion	1.0929	0.0821	0.7623	0.7722	0.5775
cn	Textualised Feature	0.9750	0.2000	0.8283	0.8362	0.6459

Table 1: Main development-set results [dev] for the closed track. Δ RMSE is computed as $\text{RMSE}_{\text{baseline}} - \text{RMSE}_{\text{system}}$, so higher values indicate larger improvements. best, near-best, close to baseline.

development-set result for each language. The modelling strategy also has a clear effect. Late Fusion performs worse than both Textualised Feature Regression and Neural Fusion, indicating that simply combining frozen sentence embeddings with classical regressors is not the most effective way to use these representations. In contrast, Neural Fusion benefits from combining encoder representations with engineered features in a more integrated way. Since the best development results are obtained with Neural Fusion using BGE-M3 or multilingual-E5-large, we use this setting for the subsequent feature-ablation and feature-importance analyses.

4.1 Feature importance and ablations

We analyse the engineered features in three steps. First, we compute Kendall’s τ correlation between each feature and the gold GLMM score for each language. Second, we use these rankings to construct two compact feature sets: Top-1 per feature family, which keeps the single highest-ranked feature in each family, and Top-3 per feature family, which keeps the three highest-ranked features in each family. Third, we compare these compact settings with models trained using only one feature family at a time. This separates feature–target association

from the predictive value of compact subsets and single feature families.

The Kendall rankings show that frequency and lexical familiarity are the most consistent individual signals. In Spanish, the top-ranked features are subtitle-based target frequency ($|\tau| = 0.2730$), the retrieval prior combining clue/context support and candidate-space size ($|\tau| = 0.2643$), and subtitle contextual diversity ($|\tau| = 0.2572$). In German, the retrieval prior ranks first ($|\tau| = 0.2985$), followed by subtitle frequency ($|\tau| = 0.2837$) and subtitle contextual diversity ($|\tau| = 0.2680$). In Chinese, the strongest associations are subtitle frequency ($|\tau| = 0.4128$), subtitle contextual diversity ($|\tau| = 0.4127$), and the KELLY lexical familiarity score ($|\tau| = 0.4029$).

Table 2 shows that that frequency-based features (subtitle frequency and contextual diversity) emerge as the most consistent and strongest individual predictors across languages, but its role differs by L1. Spanish is the clearest frequency-driven case: the frequency-only model gives the best Dev RMSE, 1.0855, slightly ahead of all features, 1.0952, with nearly identical OOF RMSE, 1.0987 versus 1.0988. German shows a more balanced pattern, where surprisal gives the best Dev

Language	Feature setting	OOF				Dev			
		RMSE	Pearson	Spearman	Kendall	RMSE	Pearson	Spearman	Kendall
es	All features	1.0988	0.8172	0.8149	0.6223	1.0952	0.8324	0.8373	0.6473
	Top-1/group (Kendall)	1.1429	0.7931	0.7900	0.5954	1.1256	0.8244	0.8298	0.6378
	Top-3/group (Kendall)	1.1221	0.8061	0.8036	0.6089	1.1143	0.8308	0.8347	0.6452
	Frequency	1.0987	0.8122	0.8084	0.6157	1.0855	0.8320	0.8350	0.6450
	Surface-form	1.1343	0.8036	0.8013	0.6076	1.1739	0.8216	0.8238	0.6315
	MLM	1.2928	0.7257	0.7174	0.5274	1.1099	0.8295	0.8297	0.6404
	Surprisal	1.1268	0.7997	0.7966	0.6022	1.0973	0.8304	0.8354	0.6440
	Semantic	1.1276	0.8043	0.8029	0.6099	1.1209	0.8269	0.8275	0.6381
	Cognate	1.1281	0.8059	0.8036	0.6104	1.1235	0.8237	0.8289	0.6386
	Features only	1.3248	0.7027	0.6927	0.5030	1.3274	0.7136	0.7079	0.5130
de	All features	1.0761	0.7987	0.8036	0.6080	1.0873	0.8234	0.8414	0.6446
	Top-1/group (Kendall)	1.0538	0.8210	0.8254	0.6313	1.1056	0.8250	0.8398	0.6428
	Top-3/group (Kendall)	1.0476	0.8096	0.8135	0.6189	1.1306	0.8123	0.8277	0.6304
	Frequency	1.0452	0.8147	0.8201	0.6256	1.0941	0.8235	0.8381	0.6413
	Surface-form	1.0805	0.8060	0.8125	0.6168	1.1073	0.8173	0.8335	0.6361
	MLM	1.0778	0.8028	0.8088	0.6129	1.1255	0.8132	0.8280	0.6285
	Surprisal	1.0831	0.7963	0.8025	0.6060	1.0861	0.8261	0.8403	0.6454
	Semantic	1.0744	0.7975	0.8036	0.6075	1.1343	0.8140	0.8300	0.6334
	Cognate	1.0806	0.8083	0.8133	0.6190	1.0985	0.8225	0.8389	0.6431
	Features only	1.2491	0.7086	0.7080	0.5146	1.3339	0.6825	0.6893	0.4986
cn	All features	0.9594	0.8235	0.8200	0.6300	0.9681	0.8351	0.8428	0.6568
	Top-1/group	0.9951	0.8258	0.8365	0.6493	0.9844	0.8139	0.8134	0.6215
	Top-3/group	0.9651	0.8278	0.8231	0.6330	1.0126	0.8217	0.8334	0.6456
	Frequency	0.9492	0.8314	0.8298	0.6397	0.9899	0.8293	0.8355	0.6472
	Surface-form	0.9857	0.8173	0.8165	0.6245	0.9840	0.8298	0.8419	0.6498
	MLM	0.9649	0.8218	0.8205	0.6305	0.9796	0.8280	0.8403	0.6498
	Surprisal	0.9689	0.8225	0.8218	0.6305	1.0089	0.8231	0.8351	0.6460
	Semantic	1.0053	0.8050	0.8025	0.6102	1.0030	0.8261	0.8335	0.6433
	Cognate	0.9693	0.8186	0.8164	0.6256	0.9807	0.8264	0.8357	0.6463
	Features only	1.1647	0.7159	0.7080	0.5190	1.2377	0.6823	0.6925	0.5081

Table 2: Feature ablation results for Spanish, German, and Chinese. Top-1/group and Top-3/group are built by selecting the highest-ranked one or three features within each feature group according to Kendall correlation with the training GLMM score. **best**, **near-best**, **worst**.

RMSE, 1.0861, narrowly ahead of all features, 1.0873, while frequency, 1.0941, and cognate/form-similarity, 1.0985, remain close. Chinese has the most distributed signal: all features perform best, with Dev RMSE 0.9681, but MLM, cognate/form-similarity, lexical/surface form, and frequency all remain competitive.

The official test results in Table 3 partly confirm this pattern. The best UOL@IDEM runs achieve RMSE 1.132 for Spanish, 1.037 for German, and 0.891 for Chinese. Spanish is best with the all-features run, while German and Chinese are best with the frequency-oriented run. Thus, although richer feature combinations remain useful during development, frequency-based predictors generalise most robustly on the hidden test set for German and Chinese.

4.2 Error analysis

The feature-importance and ablation results show that in addition to the baseline sentence encoder, the models learn a strong difficulty signal, especially from frequency and lexical familiarity. However, a strong ranking does not guarantee good calibration. We therefore examine where frequency becomes insufficient, whether predictions are compressed across the difficulty scale, and which feature patterns distinguish low-error from high-error cases.

We first examine calibration by dividing the development items into five quantile-based gold-difficulty bands. Band 1 contains the easiest items and Band 5 the hardest items. For each language and band, we compute RMSE, MAE, Kendall’s τ , and signed bias, defined as $\hat{y} - y$, where positive val-

Language	Run	RMSE ↓	Pearson ↑
es	All features	1.132	0.813
	Frequency	1.134	0.808
	Top-3/family	1.140	0.813
de	All features	1.079	0.819
	Frequency	1.037	0.834
	Surprisal	1.078	0.818
cn	All features	0.930	0.856
	Frequency	0.891	0.860
	Top-1/family	0.919	0.858

Table 3: Official closed-track test results for UOL@IDEM submissions. Green highlights the best run.

ues indicate overprediction of difficulty (Kendall, 1938). Uncertainty is estimated using bootstrap resampling (Efron and Tibshirani, 1994).

Figure 1 shows that the language-specific neural fusion models broadly follow the gold difficulty scale, but with a clear compression toward the centre. This compression is visible in the solid mean-bias line, where bias is defined as predicted difficulty minus gold difficulty. In Band 1, which contains the easiest items, the bias is strongly positive: +1.21 for German, +1.28 for Spanish, and +1.03 for Chinese. This means that genuinely easy items are predicted as more difficult than they are. At the other end of the scale, Band 5 shows the opposite pattern: the bias is negative, with values of −0.30, −0.32, and −0.27, respectively, indicating that the hardest items are predicted as slightly easier than their gold scores. These two effects show that predictions are pulled away from the extremes and toward the middle of the difficulty scale. The same pattern is also reflected in the mean difficulty lines: the predicted mean difficulty line changes less sharply across bands than the gold mean difficulty line. Thus, the main weakness is not a failure to rank items by difficulty, but a calibration problem in which the model underestimates the full spread of the difficulty scale. The full error-analysis materials are available online.³

³<https://tinyurl.com/UoL-BEA-Error-Analysis>

Lang.	RMSE	MAE	Bias	τ	Band bias pattern
de	1.117	0.850	+0.346	0.629	+1.21 → −0.30
es	1.111	0.834	+0.331	0.651	+1.28 → −0.32
zh	0.975	0.724	+0.258	0.648	+1.03 → −0.27

Table 4: Overall development-set error profile. Bias is the mean signed error $\hat{y} - y$. The final column summarises the change from Band 1 to Band 5, showing scale compression in all three languages.

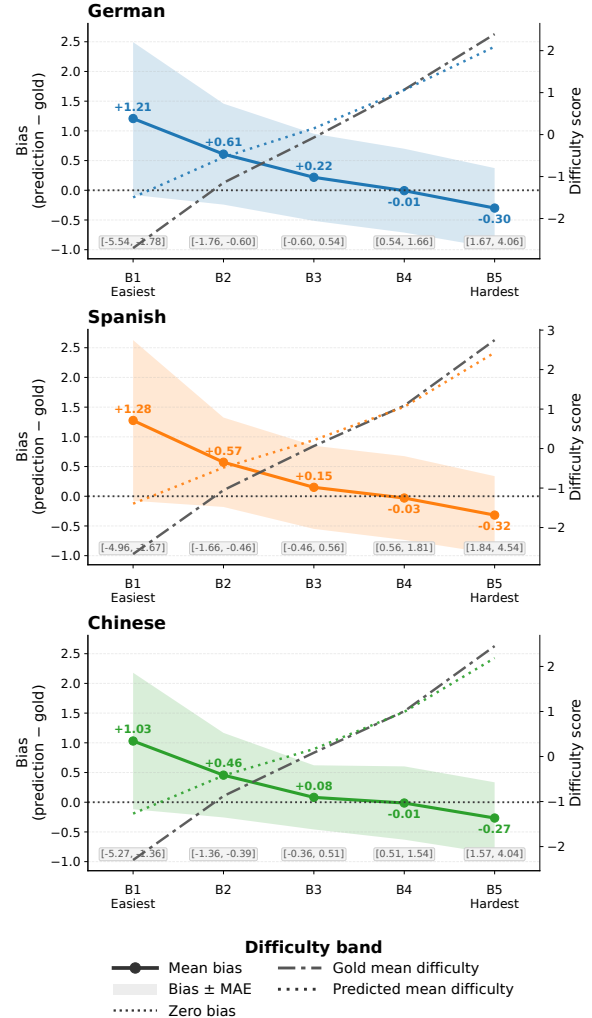


Figure 1: Calibration by gold difficulty band for German, Spanish, and Chinese. Bands B1–B5 are ordered from easiest to hardest according to the gold difficulty score. The solid line shows mean prediction bias, defined as predicted difficulty minus gold difficulty, and the shaded region shows bias $\pm$ MAE. Positive bias in the easiest bands indicates over-prediction of difficulty, while negative bias in the hardest band indicates under-prediction.

Table 4 summarises the same pattern at language level. Chinese has the lowest RMSE and MAE, but it still shows the same movement from overprediction in the easiest band to mild underprediction in the hardest band. This connects directly to the ablation results: frequency provides a robust global

difficulty signal, but it can also pull predictions toward an average level of difficulty rather than fully adapting to the extremes of the learner-specific scale. Absolute-error associations with individual features are weak: most Kendall correlations are below $|\tau| = 0.05$. The strongest cue is prediction entropy, `pred_ent`, with $\tau = 0.048$, followed by MLM entropy, learner-frequency percentile, target length, semantic-domain entropy, prediction margin, and cognate similarity. Residual errors therefore appear when several cues diverge, such as frequency, contextual predictability, semantic ambiguity, form similarity, and L1–English transparency.

To examine these interactions more directly, we compare low-error and high-error items within the same language and difficulty band. This controls for broad gold difficulty: we are not simply comparing easy words with hard words, but asking what distinguishes better and worse predictions among items of comparable true difficulty.

Within each language–band group, we define low-error items as the lowest-error quartile and high-error items as the highest-error quartile. Since many features are skewed, ordinal, or tied, we use Mann–Whitney tests and Cliff’s δ to compare feature distributions, and apply Benjamini–Hochberg correction to control the false discovery rate (Mann and Whitney, 1947; Cliff, 1993; Benjamini and Hochberg, 1995). Figure 2 reports the standardised contrast between the two groups, computed as low-error minus high-error. Positive values indicate features that are higher in better-predicted items; negative values indicate features that are higher in worse-predicted items.

The controlled contrasts clarify the source of the calibration errors. Low-error items are usually those where several cues point in the same direction: frequency, lexical form, semantic evidence, and contextual predictability jointly support the predicted difficulty. High-error items arise when these cues conflict, so the model receives mixed evidence about whether the item should be easy or difficult.

In German, low-error items show stronger form and contextual-predictability signals, including target length, source length, target syllable count, MLM rank, and surprisal features. High-error items are more associated with subtitle frequency, subtitle contextual diversity, and cognate similarity. This pattern indicates that German predictions are most stable when frequency is supported by context-

tual and form-based evidence, but less stable when frequency or surface similarity dominates without enough contextual support.

Spanish shows the clearest example of this conflict. The most error-prone cell is Spanish Band 1, where genuinely easy items are often overpredicted. Within this band, the strongest descriptive contrasts involve subword surprisal, MLM log probability, and pseudo-log-likelihood surprisal. The largest contrast is for `surp_sub_sum`, with Cliff’s $\delta = -0.298$ and a raw Mann–Whitney $p = 0.031$; however, this effect does not remain significant after Benjamini–Hochberg correction ($p_{BH} = 0.522$), so we treat it as exploratory. The pattern is still informative: some easy items may be frequent, familiar, cognate-supported, or transparent for Spanish-speaking learners even when the immediate context does not make them highly predictable to an MLM-style model. The model then overpredicts difficulty because contextual predictability conflicts with learner-oriented familiarity.

Chinese has the strongest overall performance, but it follows the same mechanism. Low-error items combine form, semantic-shift, cognate-similarity, and probability-based support, including target length, target syllables, semantic shift, predicted log probability, cognate similarity, and MLM log probability. High-error items are more strongly associated with frequency features such as word-frequency value, subtitle contextual diversity, subtitle log-frequency, and subtitle frequency. Thus, even for Chinese, predictions are most reliable when frequency is reinforced by semantic and form-based evidence rather than operating as the main cue alone.

Overall, the error analysis refines the feature-ablation findings. Frequency is the strongest global cue and explains much of the high ranking performance, but it does not guarantee calibrated predictions. This explains why high Kendall correlations can coexist with systematic calibration errors: the models often rank items correctly, but their predicted scores are compressed toward the middle, leading to overprediction for easy items and underprediction for hard items.

5 Conclusion

This paper presented UOL@IDEM, a closed-track system for L1-aware English vocabulary difficulty prediction. The system combines multilingual con-

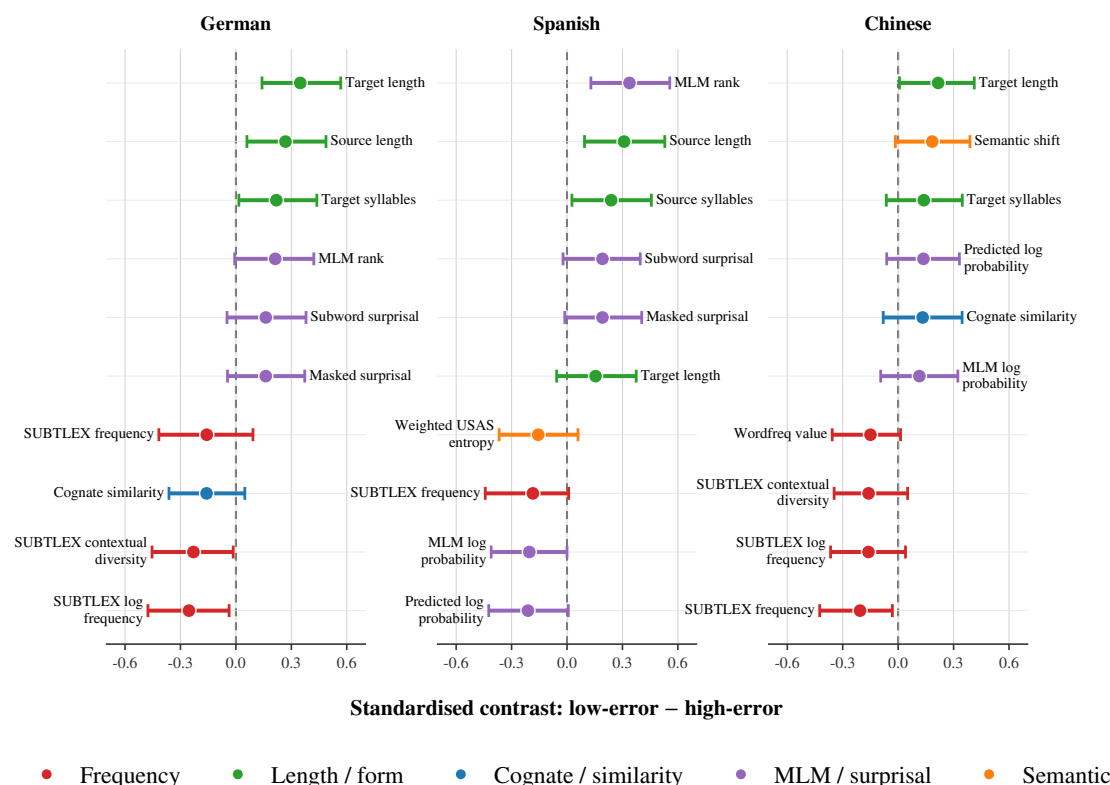


Figure 2: Matched low-error versus high-error contrasts within language and difficulty band. Positive values indicate features higher in low-error items; negative values indicate features higher in high-error items.

textual representations with engineered features capturing frequency, lexical surface form, surprisal, retrieval evidence, cognate-like similarity, and semantic alignment.

Across the development set, neural fusion with sentence-embedding encoders gives the strongest performance, improving over the closed-track base-lines by 0.2618 RMSE for Spanish, 0.2407 for German, and 0.2069 for Chinese. On the official test set, the best submissions achieve RMSE scores of 1.132, 1.037, and 0.891 for Spanish, German, and Chinese, respectively.

The analysis shows that explicit linguistic features remain useful even with strong multilingual encoders. Frequency is the most stable signal, while form, surprisal, retrieval, cognate, and semantic features provide complementary evidence for L1-aware prediction. At the same time, the error analysis shows that strong ranking performance does not always translate into well-calibrated difficulty estimates, especially at the easiest and hardest ends of the scale.

Future work should focus on calibration and on more compact models that retain the most informative frequency, form, surprisal, and semantic cues while better modelling L1-specific transfer

and contextual recoverability.

6 Limitations

Our work is limited to the closed-track shared-task data, so we do not use external learner corpora, larger lexical resources, multilingual joint training, or large language models. These could be useful for rare targets, noisy source-word entries, and ambiguous source–target mappings.

The engineered features improve interpretability but add pipeline complexity, including pre-processing, frequency lookup, retrieval scoring, surprisal estimation, semantic tagging, and cognate/similarity scoring. Following the closed-track requirements, we train separate models for each L1, which does not explicitly model shared versus language-specific difficulty signals.

The main evaluation limitation is calibration. The models overpredict the easiest items and slightly underpredict the hardest ones, indicating regression toward the middle of the difficulty scale. This analysis is restricted to the development set because gold labels for the hidden test set are not available.

Ethics Statement

This work uses data released for research under the shared-task conditions. The task concerns educational assessment and therefore has potential downstream implications for fairness and learner support. We do not claim that model predictions should replace psychometric validation or expert judgment; rather, they should be used as supportive signals in item development and educational NLP pipelines.

Acknowledgments

We thank the BEA 2026 shared-task organisers for releasing the dataset and the evaluation framework. This document is part of a project that has received funding from the European Union’s Horizon Europe research and innovation program under Grant Agreement No. 101132431 (iDEM Project). The University of Leeds was funded by UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee (Grant Agreement No. 10103529). The views and opinions expressed in this document are solely those of the author(s) and do not necessarily reflect the views of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

- Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. 2019. [Cognet: A large-scale cognate database](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3136–3145, Florence, Italy. Association for Computational Linguistics.
- Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. 2022. [A large and evolving cognate database](#). *Language Resources and Evaluation*, 56:1407–1437.
- 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.
- British Council. 2026. [Bea 2026 shared task: L1-aware vocabulary difficulty prediction](#). <https://www.britishcouncil.org/data-science-and-insights/bea2026st>. Accessed 2026-04-04.
- Marc Brysbaert and Boris New. 2009. [Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english](#). *Behavior Research Methods*, 41(4):977–990.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Norman Cliff. 1993. [Dominance statistics: Ordinal analyses to answer ordinal questions](#). *Psychological Bulletin*, 114(3):494–509.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Council of Europe. 2001. [Common European Framework of Reference for Languages: Learning, Teaching, Assessment](#). Cambridge University Press.
- 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.
- Bradley Efron and Robert J. Tibshirani. 1994. [An Introduction to the Bootstrap](#). Chapman and Hall/CRC, New York.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the BEA 2026 shared task on vocabulary difficulty prediction for english learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Ariavazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spacy: Industrial-strength natural language processing in python](#).

- M. G. Kendall. 1938. [A new measure of rank correlation](#). *Biometrika*, 30(1/2):81–93.
- Adam Kilgariff, Frieda Charalabopoulou, Maria Gavrilidou, Janne Bondi Johannessen, Saussan Khalil, Sofie Johansson Kokkinakis, Robert Lew, Serge Sharoff, Ravikiran Vadlapudi, and Elena Volodina. 2014. [Corpus-based vocabulary lists for language learners for nine languages](#). *Language Resources and Evaluation*, 48:121–163.
- Henry B. Mann and Donald 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.
- Kai North, Matthew Shardlow, Marcos Zampieri, and 1 others. 2023. [Lexical complexity prediction: An overview](#). *arXiv preprint arXiv:2303.04851*.
- Adaeze Ohuoba, Serge Sharoff, and Callum Walker. 2024. [Quantifying the contribution of MWEs and polysemy in translation errors for English–Igbo MT](#). In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 537–547, Sheffield, UK. European Association for Machine Translation (EAMT).
- Paul Rayson, Dawn Archer, Scott Piao, and Tony McEnery. 2004. [The ucrel semantic analysis system](#). In *Proceedings of the Workshop on Beyond Named Entity Recognition: Semantic Labelling for NLP Tasks in Association with the 4th International Conference on Language Resources and Evaluation (LREC 2004)*.
- Miguel Rios and Serge Sharoff. 2016. [Language adaptation for extending post-editing estimates for closely related languages](#). *The Prague Bulletin of Mathematical Linguistics*, 106:5–18.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. [Masked language model scoring](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–7, 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):1037–1064.
- Serge Sharoff, Dirk Goldhahn, and Uwe Quasthoff. 2017. *Frequency Dictionary: Russian*, volume 9 of *Frequency Dictionaries*, chapter Corpus. Leipziger Universitätsverlag. Uwe Quasthoff, Sabine Fiedler, Erla Hallsteindóttir (editors).
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. [Luminosinsight/wordfreq: v2.2](#).
- Mark J. van der Laan, Eric C. Polley, and Alan E. Hubbard. 2007. [Super learner](#). *Statistical Applications in Genetics and Molecular Biology*, 6(1):1–23.
- Walter J. B. van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. [Subtlex-uk: A new and improved word frequency database for british english](#). *The Quarterly Journal of Experimental Psychology*, 67(6):1176–1190.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. [Multilingual E5 text embeddings: A technical report](#). *arXiv preprint arXiv:2402.05672*.

A Feature Inventory

This appendix summarises the engineered features used in our experiments. Feature names follow the column names used in the analysis files and figures. For readability, features are grouped by functional role rather than presented as a single flat inventory. Table 5 gives the full engineered-feature inventory.

Table 5: Engineered features grouped by functional role.

Feature	Description
Frequency	
freq_en	English frequency estimate for the target word.
freq_en_pct_rank	Percentile rank of the English target frequency.
sub_wf	SUBTLEX word-frequency measure.
sub_lg10wf	Log-scaled SUBTLEX word-frequency measure.
sub_cd	SUBTLEX contextual-diversity measure.
sub_lg10cd	Log-scaled SUBTLEX contextual-diversity measure.
kelly_rank	Rank of the target word in a learner-oriented lexical frequency resource.
kelly_pct	Percentile rank derived from the Kelly learner-frequency list.
kelly_points	Difficulty-oriented lexical score from the Kelly list.
wf_zipf	Zipf-scaled word-frequency score for the English target word.
wf_value	Raw or transformed corpus-based word-frequency value.
Lexical, clue, and preprocessing	
tgt_len	Number of characters in the English target word.
src_len	Number of characters in the cleaned L1 source word.
tgt_syll	Syllable count of the English target word.
target_pos	Part-of-speech information for the English target word.
source_pos	Part-of-speech information for the L1 source word, where available.
clue_len	Length of the target clue.
clue_hidden_chars	Number of hidden characters in the target clue.
clue_hidden_ratio	Proportion of hidden characters in the target clue.
L1_source_word_raw	Original unnormalised source-word field preserved during preprocessing.
L1_source_word_excluded_word	Word appearing inside an exclusion note in the raw source-word field.
L1_source_word_has_excluded_word	Indicator that the source-word field contains an exclusion note.
L1_source_word_excluded_note_type	Type of exclusion note detected during preprocessing.
L1_source_word_has_alternative	Indicator that multiple source-word candidates are present.
L1_source_word_alternative_count	Number of alternative source-word candidates detected.
L1_source_word_alternatives	Preserved list of source-word alternatives extracted during preprocessing.
Retrieval	
gold_rank	Rank position of the gold English target in the candidate list after retrieval and reranking.
gold_score	Final reranked score assigned to the gold target.
top1_score	Score of the highest-ranked retrieved candidate.

Continued on next page

Table 5 – continued from previous page

Feature	Description
top2_score	Score of the second-ranked retrieved candidate.
margin12	Difference between the top-1 and top-2 candidate scores.
gold_cos	Cosine similarity between the gold target and the retrieved representation.
gold_prob	Probability-like score associated with the gold target under retrieval/reranking.
nn_pred_cos	Cosine similarity associated with the nearest-neighbour prediction.
nn_pred_prob	Probability-like confidence score for the nearest-neighbour prediction.
nn_entropy	Entropy of the retrieved candidate distribution.
cand_count	Number of valid candidate targets considered after filtering.
retrieval_target_prior	Prior retrieval score for the target independent of full contextual reranking.
retrieval_target_in_context	Context-sensitive retrieval score for the target.
baseline_pred_matches_target	Indicator that the retrieval baseline predicts the correct target.
baseline_clue_overlap	Degree of overlap between the predicted candidate and the target clue.
baseline_pred_len	Length-based compatibility between the retrieved candidate and the target constraint.
MLM and surprisal	
mlm_logp	Mean masked-language-model log probability assigned to the gold target wordpieces.
mlm_rank	Rank of the gold target under the masked-language-model prediction.
mlm_ent	Entropy of the masked-token distribution.
mlm_top1_match	Indicator that the top MLM prediction matches the gold target.
pred_logp	Log probability of the top predicted token or predicted target sequence.
pred_ent	Entropy associated with the prediction distribution.
pred_margin	Margin between the best and second-best MLM predictions.
mlm_target_pred_cos	Cosine similarity between gold-target and predicted-token embeddings.
surp_masked	Surprisal computed from simultaneous masking of all target wordpieces.
surp_pll	Surprisal computed using pseudo-log-likelihood with one masked wordpiece at a time.
surp_chain	Surprisal computed sequentially via a chain-rule decomposition over target subwords.
surp_sub_sum	Sum of surprisal contributions across target subwords.
surp_sub_mean	Mean surprisal across target subwords.
Semantic and domain shift	
usas_domain_match	Indicator of semantic-domain compatibility between source and target.

Continued on next page

Table 5 – continued from previous page

Feature	Description
usas_ent_unw	Unweighted entropy over possible semantic-domain assignments.
usas_ent_wtd	Weighted entropy over semantic-domain assignments.
sem_shift	Soft semantic-shift score measuring divergence between the L1 source word and the English target.
semantic_domain_overlap	Degree of overlap between source- and target-side semantic domains.
Cognate and cross-lingual form	
lev_sim	Weighted Levenshtein-based similarity between the L1 source word and the English target.
cog_sim	Cross-lingual form-similarity score used as a cognate or transparency cue.
char_ngram_sim	Character n-gram similarity between the L1 source word and the English target, where available.
cognet_match	Indicator or score derived from external cognate-link evidence, where available.

Table 5: Engineered features grouped by functional role.

B Neural-fusion model settings

Category	Setting	Value
Input and model	Text fields	L1_context, L1_source_word, en_target_word, optional en_target_clue
	Fusion model	Enhanced neural fusion
	Encoder backbones	XLM-R, mBERT, LaBSE, multilingual-E5, BGE-M3, and related multilingual encoders
	Text pooling	Attention pooling or CLS pooling, depending on the model configuration
Architecture	Tabular branch	Feed-forward network with configurable hidden layers, dropout, and optional residual connections
	Prediction head	MLP regression head over the concatenated text and tabular representations
Optimisation	Optimiser	AdamW
	Encoder learning rate	2×10^{-5}
	Task-head learning rate	10^{-3}
	Weight decay	0.01
	Warmup ratio	0.1
	Schedule	Linear warmup and decay
	Gradient clipping	Maximum norm 1.0
	Mixed precision	Used when GPU support is available
	Checkpoint selection	Validation RMSE
Training protocol	Epochs	5
	Training batch size	16
	Evaluation batch size	16 or 32, depending on the experiment configuration
	Early stopping	Patience of 2 epochs
Evaluation	Cross-validation	5-fold
	Main metric	RMSE
	Other metrics	Pearson, Spearman, Kendall's τ

Table 6: Model, training, and optimisation settings used in the neural-fusion experiments.

The Huber loss used in some optimisation settings is defined in Eq. 8.

$$L_{\delta}(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \leq \delta, \\ \delta \left(|a| - \frac{1}{2}\delta \right) & \text{otherwise.} \end{cases} \quad (8)$$

C USAS semantic-shift examples

Example	Source USAS tags and categories	Target USAS tags and categories	Soft sim.	Soft shift
Operation: exact tag overlap				
<i>firme–solid</i>	O4.1 O4.5 <i>General appearance and physical properties; Texture</i>	A5.1+ A5.3+ O1.1 O4.5 T2++ <i>Evaluation: Good/bad; Evaluation: Accuracy; Substances and materials: Solid; Texture; Time: Beginning and ending</i>	0.67	0.33
<i>riesgo–risk</i>	A1.4 A15– I1 I2 I2.1.3 <i>Chance/luck; Safety/Danger; Money generally; Business; Business: Generally</i>	A1.4 A15– <i>Chance/luck; Safety/Danger</i>	0.78	0.22
Operation: same fine-grained prefix				
<i>oyente–listener</i>	S2.1 S2.2 S5 X2.3 <i>People: Female; People: Male; Groups and affiliation; Learn</i>	S2mf X3.2 <i>People; Sensory: Sound</i>	0.69	0.31
<i>noticias–news</i>	Q1 Q4 <i>Linguistic actions, states and processes; The Media</i>	Q4.2 Q4.3 T3– X2.2+ <i>The Media: Newspapers; The Media: TV, radio and cinema; Time: Old/new/young; Knowledge</i>	0.60	0.40
Operation: same major semantic field				
<i>ladrillo–brick</i>	O1 O4.3 O4.4 <i>Substances and materials generally; Colour and colour patterns; Shape</i>	H1 O2 S8+ <i>Architecture and kinds of houses/buildings; Objects generally; Helping/hindering</i>	0.43	0.57
<i>debatir–discuss</i>	A6.1 G1.2 Q1.1 <i>Comparing: Similar/different; Politics; Linguistic actions and communication</i>	Q2.1 Q2.2 <i>Speech: Communicative; Speech acts</i>	0.42	0.58
Operation: related major semantic fields				
<i>sueño–dream</i>	B3 T1.3 <i>Medicines and medical treatment; Time: Period</i>	A5.2– X4.1 X7+ <i>Evaluation: True/false; Mental object: Conceptual object; Wanting/planning/choosing</i>	0.18	0.82
<i>obra–comedy</i>	A1.1.1 I3.1 <i>General actions, making etc.; Work and employment: Generally</i>	E4.1+ K4 Q4.3 <i>Happy/sad: Happy; Drama, theatre and showbusiness; The Media: TV, radio and cinema</i>	0.20	0.80
Operation: unrelated major semantic fields				
<i>fuegos–fireworks</i>	L1 O1 O4.6 <i>Life and living things; Substances and materials generally; Temperature</i>	G3 K1 <i>Warfare, defence and weapons; Entertainment generally</i>	0.05	0.95
<i>circulación–traffic</i>	O4.4 <i>Shape</i>	G2.1– I2.2 M1 M3 M4 S1.1.2+ <i>Crime, law and order; Business: Selling; Moving/coming/going; Vehicles and transport on land; Shipping/swimming; Reciprocity</i>	0.05	0.95
Operation: coverage gap				
<i>extensión–spread</i>	Z99 <i>Unmatched</i>	A2.1 A6.3+ F1 N3.3 N3.6 N5+ Q4.2 <i>Affect: Modify/change; Comparing: Variety; Food; Measurement: Distance; Measurement: Area; Quantities; The Media: Newspapers</i>	–	0.85 [†]

Table 7: Spanish examples of the soft USAS semantic-shift measure using richer source and target tag sets. Each cell reports the full available USAS tag set, with the corresponding category names underneath. The operation row describes the strongest source–target tag-pair relation used to characterise each example group, while the soft similarity and soft shift are computed over the full source and target tag sets. Therefore, an item can contain an exact overlapping tag but still receive a non-zero shift score when the remaining tags differ. [†]Coverage-gap scores are penalty values rather than observed semantic-distance scores.

Sakura at BEA 2026 Shared Task 1: What Makes Vocabulary Difficult?

Adam Nohejl¹ Xuanxin Wu² Yusuke Ide³
Maria Angelica Riera Machin³ Yi-Ning Chang⁴ Hitomi Yanaka^{1,5,6}

¹RIKEN ²The University of Osaka ³Nara Institute of Science and Technology
⁴National Tsing Hua University ⁵The University of Tokyo ⁶Tohoku University
adam.nohejl@riken.jp xuanxin.wu@ist.osaka-u.ac.jp ide.yusuke.ja6@is.naist.jp
riera_machin.maria.rn9@naist.ac.jp changyn@gapp.nthu.edu.tw hyanaka@is.s.u-tokyo.ac.jp

Abstract

We describe two types of models for vocabulary difficulty prediction: a high-accuracy black-box model, which achieved the top shared task result in the open track, and an explainable model, which outperforms a fine-tuned encoder baseline. As the black-box model, we fine-tuned an LLM using a soft-target loss function for effective application to the rating task, achieving $r > 0.91$. The explainable model provides insights into what impacts the difficulty of each item while maintaining a strong correlation ($r > 0.77$). We further analyze the results, demonstrating that the difficulty of items in the British Council’s Knowledge-based Vocabulary Lists (KVL) is often affected by spelling difficulty or the construction of the test items, in addition to the genuine production difficulty of the words. We make our code available online.¹

1 Introduction

The goal of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners (Felice and Skidmore, 2026) is to build models of the difficulty of English words given a learner’s L1. Such difficulty predictions can be used as a basis for pedagogical materials or computer-adaptive tests.

The shared task utilizes a large dataset with vocabulary difficulty scores for L1 Chinese, German, and Spanish learners, spanning thousands of vocabulary test items, the British Council’s Knowledge-based Vocabulary Lists (KVL; Schmitt et al., 2021, 2024). Each test item, as shown in the example in Table 1, consists of an equivalent L1 word, L1 context, and a clue for the first letter of the English word and its length in letters.

The test format, therefore, focuses on productive knowledge, in particular, on the ability to write the English word with the correct spelling given the test prompt. The difficulty scores, which we aim to predict, are intercept values of a generalized

L1		Spanish
English word		house
Part of speech		noun
Test item	L1 word	casa
	L1 context	Vivo en una casa grande que tiene tres dormitorios.
	Clue	h _ _ _ _ first letter and blanks
Difficulty score		3.07 ↑ easy, ↓ difficult

Table 1: Example of an item in the KVL data.

linear mixed model (GLMM), i.e., the log-odds of a learner responding correctly.

The shared task consisted of a closed track and an open track. While the open track was completely unrestricted, in the closed track, the use of large language models (LLMs) was limited to feature extraction, and only the L1-specific training data could be used. We present two approaches to vocabulary difficulty modeling in this setting, each leveraging LLMs in a different way:

LLMs fine-tuned using soft targets. We propose a simple yet novel technique to fine-tune LLMs for continuous value prediction using soft targets. Models based on this technique have outperformed all other shared task submissions in the open track.

Explainable model. We build a model using well-defined features, such as similarity to the L1 word and spelling difficulty, some of which are based on LLM prompting, and use SHapley Additive exPlanations (SHAP; Lundberg and Lee, 2017) to quantify the impact of each feature on the prediction. The model performed competitively in the closed track, surpassing the fine-tuned encoder baseline.

Our analysis indicates two factors contributing to difficulty scores in the KVL data beyond the productive difficulty of words: the spelling difficulty and the choice of L1 equivalents and context in some test items.

¹<https://github.com/ynklab/vocabulary-difficulty>

2 Related Research

The task of vocabulary difficulty prediction, as construed by the BEA 2026 Shared Task (Felice and Skidmore, 2026), effectively combines lexical complexity prediction (LCP) or complex word identification (CWI) with test item difficulty estimation.

CWI is the task of identifying complex words in a sentence context. CWI shared tasks (Paetzold and Specia, 2016; Yimam et al., 2018) were dominated by feature-based systems focusing on word-level features.

LCP is an extension of CWI, where complexity is predicted on a continuous scale. The best performing approaches in an early shared task (Shardlow et al., 2021) were based on fine-tuning masked language models (MLMs). In the BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline (MLSP; Shardlow et al., 2024), where lexical complexity was predicted for ten languages with limited training data, the top-performing systems were based either exclusively on LLM prompting or on word-level features.

Nohejl et al. (2025a) proposed an LLM prompting-based LCP method, G-SCALE, which aligns with the desired scale by applying temperature scaling to the probabilities and a linear regression to the final output. They also demonstrated that the addition of a single feature, log-frequency, can further improve the LLM-based predictions.

Smădu et al. (2024) compared feature-based models and fine-tuned Transformers (MLMs and LLMs) on CWI and LCP datasets with training data consisting of thousands of examples, i.e., in a setting similar to the present shared task. The study concluded that the LLMs rarely outperform the computationally less demanding MLMs and feature-based models. Smădu et al. (2024) fine-tuned LLMs only using the standard cross-entropy loss with discretized complexity values as hard targets. We achieve large improvements in LLM and MLM fine-tuning performance by using soft targets to sidestep discretization.

Cross-entropy loss with soft targets is often associated with knowledge distillation (Hinton et al., 2015) and has been used for distilling MLMs and LLMs (e.g., Sanh et al., 2019). To the best of our knowledge, however, it has not been used for fine-tuning to predict continuous values.

Similar to LCP and CWI, the goal of the vocabulary difficulty prediction task is to estimate the difficulty of words given a specific context that de-

termines their sense. There are, however, prominent differences:

1. The complexity in LCP and CWI is measured by *subjective ratings*. The difficulty in the present task is based on the *success rate of test items*.
2. The input for LCP and CWI is only a word and its context, both in the same language. The input for this task consists of multiple elements in English and in L1.
3. LCP and CWI focus on *reading comprehension*. The present task focuses on *written production*.

We address the specific aspects of this task while drawing on insights from prior research on LCP and CWI.

Skidmore et al. (2025) fine-tuned encoders (i.e., MLMs) on the KVL data. Their fine-tuned XLM-RoBERTa model serves as the baseline of the shared task. They used SHAP as a tool for error analysis, attributing inaccurate predictions to specific input token positions. We use SHAP for explainability, attributing predictions to higher-level features such as spelling difficulty or similarity to L2.

3 Method

We propose two core methods: fine-tuning LLMs and MLMs with soft targets to predict the continuous difficulty values, and an explainable model with LLM-extracted features. To further improve the accuracy of both methods, we experiment with ensembling and additional features.

3.1 Fine-Tuning with Soft Targets

In the following, we assume an LLM or an MLM that has a token vocabulary V and predicts a probability distribution of tokens $i \in V$ conditioned on an input $\mathbf{x}$, denoted as $\hat{p}(i | \mathbf{x})$.

The standard loss function for MLM and LLM training is a cross-entropy loss with a hard target, where the entire probability mass is assigned to a single target token, i.e., the negative log likelihood of the target token. Adapting such models to predict continuous values calls for a different approach.

While encoder models (typically MLMs) are sometimes fine-tuned for the prediction of continuous values using a regression head and mean squared error (MSE) loss, this is not the case for LLMs. The prevalent supervised fine-tuning (SFT)

paradigm for LLMs is to convert such tasks to text generation by discretizing the continuous values and using the aforementioned cross-entropy with hard targets

$$\ell = -\log \hat{p}(v(d(y)) \mid \mathbf{x}), \quad (1)$$

where d is the discretization (e.g., rounding to the nearest integer), and v is the mapping of discretized values to tokens.

These common approaches to predicting continuous values using MLMs and LLMs are reflected in the LCP methods based on encoders (e.g., [Ide et al., 2023](#)), in the present shared task’s encoder baseline ([Skidmore et al., 2025](#)), and in the LCP methods based on LLMs (e.g., [Smádu et al., 2024](#)). In contrast to the MSE loss used by the encoder-based regression, the standard cross-entropy loss for LLMs requires the continuous target values to be discretized into a small set of labels, losing precision in the process.

Our method sidesteps this apparent misalignment between text generation and the prediction of continuous values by fine-tuning LLMs using cross-entropy loss with soft targets

$$\ell = -\sum_{i \in V} p(i) \log \hat{p}(i \mid \mathbf{x}). \quad (2)$$

We prompt the model to predict values on a discrete scale S in the form of successive integer points, e.g., $S = \{1, 2, 3, 4, 5\}$. Our aim, however, is to predict continuous values $y \in [\min S, \max S]$. We therefore express y as a probability-weighted sum of its nearest points. Namely, we select two points, a and $a + 1$, on the scale S such that $a \leq y \leq a + 1$, and define the probability as

$$p(i) = \begin{cases} (a + 1) - y & \text{if } i = v(a), \\ y - a & \text{if } i = v(a + 1), \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

We use p as the soft target probability in the loss function defined in [Equation 2](#).

At inference time, we predict the token probability distribution $\hat{p}$ of a single token and then compute the final output $\hat{y}$ as a token probability-weighted mean:

$$\hat{y} = \frac{\sum_{s \in S} \hat{p}(v(s) \mid \mathbf{x}) \cdot s}{\sum_{s \in S} \hat{p}(v(s) \mid \mathbf{x})}. \quad (4)$$

The same inference technique was used for LLMs by [Liu et al. \(2023\)](#) to infer continuous scores using few-shot learning, and its variations have

been applied to LCP ([Aumiller and Gertz, 2022](#); [Enomoto et al., 2024](#); [Smádu et al., 2024](#); [Nohejl et al., 2025a](#)) but without the complementary loss function for fine-tuning.

Because these two complementary techniques, soft-target cross-entropy loss and probability-weighted inference, require a single token to be predicted, they can be used not only for LLMs but also for MLMs via masked token prediction. For LLMs, the input $\mathbf{x}$ is simply the prompt; for MLMs, it is the sequence “[CLS] *prompt* [MASK] [SEP]”.

3.2 Explainable Model

Directly fine-tuning of LLMs or MLMs on this task is efficient but results in a black-box model, i.e., a model whose inner decision process is difficult to interpret. As an alternative model explainable using SHAP, we train an XGBoost ([Chen and Guestrin, 2016](#)) regressor. SHAP provides an explanation of the model’s output $f(\mathbf{x})$ via SHAP values ϕ_i that additively express the local importance of each feature x_i :

$$f(\mathbf{x}) = E[f(\mathbf{x})] + \sum_{i=1}^n \phi_i. \quad (5)$$

The SHAP values can be positive or negative, expressing how much each feature pushes a prediction higher or lower relative to the expected value. We use the additive nature of SHAP values and express the importance of groups of related features as sums of their SHAP values.

Our explainable model uses the following features:

Production frequency. We use the log-frequency of the English word in the Lang-8 learner corpus ([Mizumoto et al., 2011](#)), to estimate its written production frequency by learners. Three features represent the subcorpora of (1) all learners, (2) L1 Spanish learners, and (3) L1 Chinese learners. There was not enough data for L1 German learners.

Reception frequency. We use the log-frequency and log-range of the English word in corpora representative of spoken language to estimate its reception frequency. Namely, we use the logarithm of (1) frequency and (2) range (YouTube channels) in TUBELEX ([Nohejl et al., 2025b](#)), and (3) frequency in the spoken subcorpus of the British National Corpus (BNC; [BNC Consortium, 2007](#)).

CEFR level. We use the minimum CEFR level of the English word from the Cambridge English Vocabulary Profile (Capel, 2012)

Word length. We measure the length of the English word in letters.

L1 similarity. We compute the character-level similarity of the English word to the L1 word based on their length-normalized Levenshtein distance after removing diacritics and lowercasing. This feature is only applicable to languages written in the alphabet, Spanish and German.

Spelling difficulty. We prompt GPT-5.2 (OpenAI, 2025) to rate the spelling difficulty of an English word for each L1, given the L1 equivalent and the English word’s pronunciation.

Lexical ambiguity. We prompt (1) GPT-5.2 and (2) DeepSeek-V3.2 (DeepSeek-AI (2025)) to determine the lexical ambiguity of the test item. A lexically ambiguous item must fulfill two conditions: (a) the English word is polysemous or one of several homonyms, and (b) the particular sense referenced by the test item is unfamiliar or challenging for learners.

L1 calque. We prompt GPT-5.2 to determine if the English word is a morpheme-for-morpheme translation of the L1 equivalent. The prompt excludes single-morpheme words and simple borrowings but does not require one word to be a calque of the other in the etymological sense. In contrast to the L1 similarity feature, L1 calque is also applicable to Chinese, e.g., 超级英雄 ‘superhero’ (composed of morphemes 超级 ‘super-’ and 英雄 ‘hero’).

For each of the prompt-based features, including the binary ones, we apply G-SCALE (Nohejl et al., 2025a), i.e., temperature-scaled softmax followed by a probability-weighted mean of the LLM’s predictions (Equation 4). The optimal temperature is determined for each model separately using cross-validation. This results in continuous feature values.

When reporting SHAP values, we sum the values within each group if it consists of multiple features (e.g., production frequency) without distinguishing the individual features.

We have submitted our explainable model as explainable and the model using only the traditional (not LLM-based) subset of features as traditional.

3.3 Ensembles and Additional Features

To further increase accuracy, we ensemble fine-tuned LLMs in a linear stack, combine fine-tuned LLM predictions with features, and experiment with additional features and fine-tuning encoder models using the same method we used for LLMs.

Linear stacking. When fine-tuning LLMs, we perform 5-fold cross-validation and use the out-of-fold predictions to fit a linear regression that combines the predictions of multiple LLMs in an ensemble. The linear regression is fit separately for each L1. The final ensemble uses models fine-tuned on the complete data (the union of the provided test and development subsets). As this approach is rather computationally intensive, we compare it with simple average ensembling. Corresponding submission: `finetuned_llms`.

Enhancing LLMs with features. The linear stacking approach allows us to add our explainable features to the same linear regression as the fine-tuned LLMs, building on the G-SCALE method proposed by (Nohejl et al., 2025a). Corresponding submission: `finetuned_llms_plus`.

Fine-tuned encoders and additional features. We add an encoder model fine-tuned on single-language data and several other features to the feature model. We do this to maximize performance within the rules for the closed track, although the resulting model is no longer explainable. The added features are: (1) frequency in OpenSubtitles (Lison et al., 2018), a corpus similar to the already included TUBELEX; (2) frequency in the written subcorpus of the BNC; (3) CEFR level in the Global Scale of English² (Jong et al., 2016), similar to the already included EVP level; (4) the Glasgow norms for concreteness, imageability, familiarity, and age of acquisition (Scott et al., 2019), which often strongly correlate with other already included features; and (5) an additional prompt for the calque feature. Corresponding submission: `closed_max`.

In the open track, we prompt two LLMs, GPT-4.1-mini and GPT-4.1-nano (OpenAI, 2024), to solve the test items, and we use their probability of a correct answer to indirectly assess the test item difficulty caused by the choice of the L1 word and context in the test item. To clearly distinguish this from the difficulty of the English vocabulary itself, we call this feature **trickiness**. We hypothesize

²<https://www.english.com/gse/teacher-toolkit/user/vocabulary>

that the LLMs have near-perfect knowledge of common vocabulary in the four relevant languages, and their performance therefore reflects the trickiness of the test items rather than the difficulty of the English vocabulary. Corresponding submission: `finetuned_llms_plus`, `open_max`.

We additionally experimented with prompting two recent LLMs, GPT-5.2 and GPT-4.1 (OpenAI, 2024), to directly rate the **difficulty** of the test items for learners of each L1 using 3-shot prompting. Corresponding submission: `open_max`.

Neither of the last two prompting approaches (trickiness and difficulty) were permitted in the closed track.

4 Experimental Settings

We experimented with multiple prompts for fine-tuning using a small LLM and 1-epoch training on a single L1. However, as we show in the ablation analysis in Section 6.1, with more training epochs, the prompt matters very little. For LLM prompting, we verified on a hand-picked sample that the model responses match our expectations, but we did not try to optimize the prompts.

In Appendix A, we provide complete listings of prompts used for fine-tuning and prompting.

As a basis for all fine-tuning experiments, we used pre-trained MLMs or LLMs, i.e., models not fine-tuned on chat or instruction data. For fine-tuning, we used recent LLMs with up to 32B parameters from the GLM-4 family (Team GLM, 2024), Qwen2.5 family (Qwen Team, 2025), and Ministral-3 family (Mistral AI, 2026), as well as MLMs mmBERT (Marone et al., 2025) and XLM-RoBERTa (Conneau et al., 2020). For prompting, we used models of the GPT-4.1 (OpenAI, 2024) and GPT-5.2 (OpenAI, 2025) families, and DeepSeek-V3.2 (DeepSeek-AI, 2025). Appendix B provides more details.

For MLMs, we performed end-to-end fine-tuning. For LLMs, we applied 4-bit quantization and fine-tuned QLoRA adapters (Detrmers et al., 2023) targeting all linear modules. All hyperparameters are described in Appendix C. We trained all models on the union of the test and development subsets provided by the shared task organizers. For the prompt-based features, we called models via the OpenAI API with zero temperature and requested log-probabilities necessary for probability weighting.

In all cases where a model is supposed to output

System	Chinese	German	Spanish	Mean
<code>open_max</code>	0.631	0.723	0.743	0.699
<code>finetuned_llms_plus</code>	0.630	0.726	0.742	0.699
<code>finetuned_llms</code>	0.640	0.731	0.760	0.710
≤32B LLM Average	0.645	0.743	0.767	0.719
- GLM-4-32B	0.678	0.769	0.805	0.751
- Qwen2.5-32B	0.678	0.777	0.799	0.752
- Ministral-3-14B	0.681	0.781	0.799	0.753
≤14B LLM Average	0.662	0.770	0.804	0.745
≤9B LLM Average	0.683	0.791	0.835	0.769
Open-Track Baseline	1.034	1.166	1.198	1.133
Statistical Optimum	0.321	0.304	0.205	0.277

Table 2: RMSE of our open-track submissions, compared with average ensembles by model size, individual models, and the shared task’s open-track baseline.

a difficulty rating on a scale from 1 to 5, we map the difficulty scores from the KVL data linearly so that the highest score (the easiest item) in the training data maps to 5 and the lowest score to 1. The prompts are formulated accordingly. As the GLMM scores represent log-odds, we also experimented with mapping them to probabilities using the expit function (logistic curve), which decreased performance. See results in Appendix G.

5 Results

In line with the shared task’s evaluation, we report the root mean square error (RMSE) by L1 and compare it with the official shared tasks’ baseline, a fine-tuned XLM-RoBERTa model (Felice and Skidmore, 2026).

Note that RMSE is in the same units as the difficulty scores, which typically range from -5 to $+5$. We also report the Pearson’s correlation coefficients (PCC) r , the secondary evaluation metric, which may be easier to interpret, in Appendix D.

To further put the results into perspective, we report a “Statistical Optimum” result, which simulates the largest error (lowest correlation) that should be considered optimal given the precision of the gold standard data. The simulation is based on confidence intervals reported by Schmitt et al. (2024); see Appendix F for details.

5.1 Open Track

As shown in Table 2, the results of our three open-track submissions are very close to each other. In summary, the results do not justify the cost of adding features to fine-tuned LLMs. This contrasts with previous findings for in-context learning setting, where just adding frequency as a feature

System	Chinese	German	Spanish	Mean
closed_max	0.816	0.963	0.983	0.921
explainable	0.920	1.126	1.156	1.067
traditional	1.078	1.195	1.305	1.193
expl. – std. inference	0.961	1.151	1.190	1.101
expl. – lin. regression	0.975	1.154	1.202	1.111
Closed-Track Baseline	1.140	1.258	1.257	1.218

Table 3: RMSE of our closed-track submissions, compared with two variants of the explainable model and the shared task’s closed-track baseline.

improves LLM predictions of lexical complexity (Nohejl et al., 2025a). The models with added features (open_max and finetuned_llms_plus) surpassed all other submissions in the open track on all three languages. The linear stack of LLMs (finetuned_llms) surpassed all other submissions on Chinese and German but was narrowly outperformed by submissions from the Glite team on Spanish. The RMSE of approximately 0.7 that our models achieve is relatively close to the statistical optimum of 0.277.

Linear stacking (finetuned_llms) results in a modest improvement over the average ensemble of the same models ($\leq 32\text{B}$ LLM Average). The ensembling itself, however, improves performance more substantially, from an RMSE of 0.751–0.753 for individual models to 0.719 for the average ensemble. While increasing the model size obviously improves performance, it is worth noting that the smaller Ministral-3-14B model performs on par with the 32B Qwen2.5 and GLM-4 models.

In Appendix E, we compare all individual LLMs used in the ensembles above, as well as LLMs with 0.5B parameters and similarly sized MLMs, showing that LLMs and MLMs achieve comparable results at comparable model sizes.

5.2 Closed Track

Table 3 compares the results of our submissions to the closed-track baseline. Interestingly, our model with only traditional features performs on par with the encoder-based baseline, and our explainable model with extra LLM-based features outperforms the baseline by a relatively wide margin (mean RMSE of 1.218 vs. 1.067). The closed_max model achieves a further improvement (mean RMSE of 0.921) by adding a fine-tuned MLM as a feature.

Two variants applied to the explainable features—standard inference for prompt-based features instead of probability weighting, and linear regression in-

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (Ministral-3-14B)	0.681	0.781	0.799	0.753
- single language	0.701	0.824	0.799	0.775
- out-of-language	0.874	0.901	0.999	0.925
- short prompt	0.697	0.790	0.808	0.765
- standard loss	0.762	0.841	0.892	0.832
- std. loss & inference	0.863	0.943	1.003	0.936

Table 4: RMSE of ablations of our LLM-based model.

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (mmBERT-b)	0.921	0.984	1.063	0.989
- single language	0.916	1.044	1.070	1.010
- out-of-language	1.115	1.102	1.193	1.136
- short prompt	0.929	1.012	1.066	1.002
- standard loss	1.003	1.099	1.166	1.089
- std. loss & inference	1.064	1.144	1.192	1.133
Regression (XLMR-b)	1.222	1.276	1.373	1.290
Regression (XLMR-l)	1.048	1.144	1.192	1.128
Reg. (mmBERT-b)	1.000	1.105	1.153	1.086

Table 5: RMSE of ablations of our MLM-based model, compared with using a standard regression head and different base models (XLM-RoBERTa-base/large).

stead of an XGBoost regressor—result in higher RMSE.

6 Analysis and Discussion

6.1 Ablation Analysis

We ablate the fine-tuned LLM and MLM that were used in our open-track submissions and the closed_max closed-track submission. For the open-track submission, we ablate only one of the three similarly performing LLMs in the ensemble, Ministral-3-14B as follows:

Single language. We fine-tune the model on single L1 data instead of all three languages.

Out-of-language. We fine-tune the model on data for the other two L1s (e.g. the model tested on Chinese is fine-tuned on German and Spanish).

Short prompt. We use a minimalistic prompt template instead of the basic verbose one. See Section A.1.

Standard loss. Instead of using the cross-entropy loss with a soft target, described in Section 3.1, we fine-tune using the standard cross-entropy loss with the discretized score following Smřdu et al. (2024).

Standard loss and inference. We fine-tune using the standard cross-entropy loss with a discretized

score, and instead of using probability-weighting at inference, we decode the highest probability token.

In addition to performing the same ablations for the MLM, we also compare the results of our method to those of using a regression head with MSE loss on mmBERT-base and two sizes of XLM-RoBERTa models (listed as **Regression**). Note that in the closed track, rules required us to use the variant we report here as an ablation (mmBERT-b/single language).

As shown in Table 4 and Table 5, our method outperforms all ablations with the single exception of the MLM model fine-tuned only on Chinese L1 data, which performs better on Chinese by a small margin. In the other cases, training on a single language results in only a small performance drop.

The standard loss ablation demonstrates that our technique of fine-tuning using cross-entropy loss with soft targets is superior to common approaches for both LLMs and MLMs. Fine-tuning using the standard loss results in a tangible performance drop for both model types (a difference in mean RMSE of 0.079 and 0.900, respectively). For the MLM, the common approach of fine-tuning a regression head with MSE loss degrades performance similarly (by a difference in mean RMSE of 0.809).

6.2 The Explainable Models

As outlined in Section 3.2, we designed a feature-based model with traditional and LLM-based features that all have a clear interpretation. SHAP values explain the model’s output locally in terms of the impact of each feature, which has both sign and magnitude. For a global analysis of feature importance, we report the mean absolute SHAP values computed over the test data in Figure 1.

Production frequency is a consistently high-impact feature across languages. Its importance per language, in fact, corresponds with the amount of L1-specific data available. Yet, even for German, with no L1-specific data available, the feature almost completely overshadows **reception frequency**, which ranks as the least important across L1s. This is in line with the focus of the KVL data on production and can be contrasted with findings for the comprehension-focused LCP, where reception frequency (represented by frequency in TUBELEX) and production frequency (represented by frequency in Lang-8) are equally good predictors (Nohejl et al., 2024)

Spelling difficulty is the most important feature for L1 German and the second most important

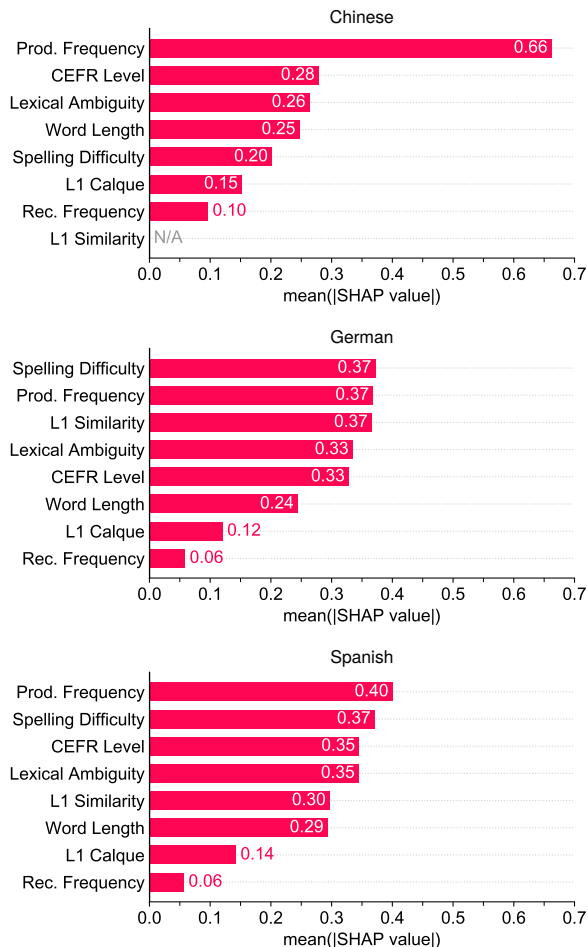


Figure 1: Global SHAP summaries by L1.

feature for Spanish. Perhaps counter-intuitively, it is much less important for L1 Chinese. We hypothesize this is caused by two factors. First, the production frequency uses learner-written texts, and therefore it partially discounts the frequency of words with frequent mistakes. As a result, the importance of the separate spelling difficulty feature is lower proportionally to the size of the L1-specific written production data. Second, for Chinese speakers, there may be less interference from the orthography of their L1, which is primarily written in Chinese characters, while Spanish and German speakers may be more prone to making errors due to the influence of their L1’s spelling (e.g., English *music, action*; German *Musik, Aktion*).

CEFR level ranks particularly high for Chinese L1 learners. This corresponds to a very small proportion of cognates between English and Chinese compared to the two European languages, where **L1 Similarity** can play a comparable (Spanish) or more important role (German).

Lexical ambiguity has a similarly high impact as

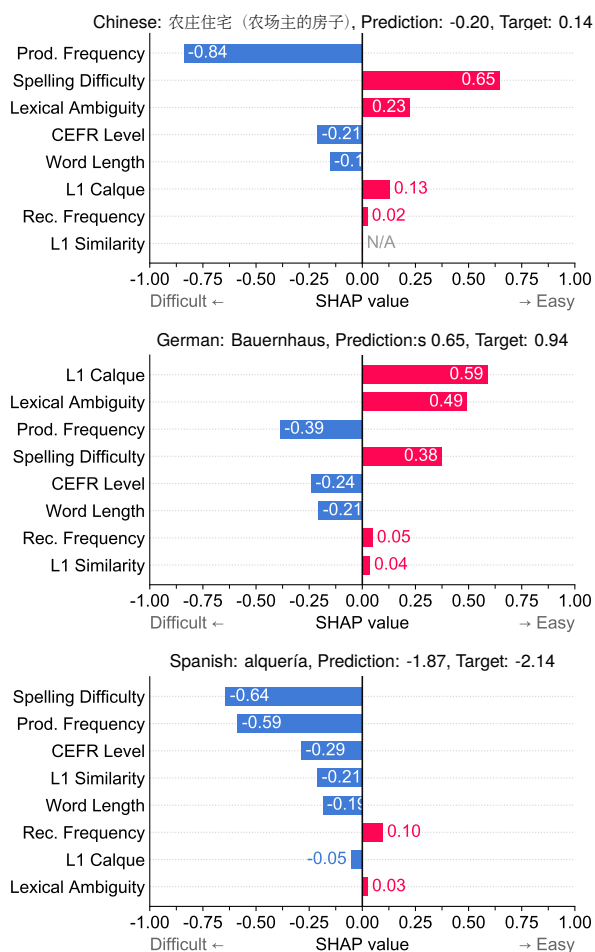


Figure 2: Example of local SHAP explanations by L1 for the English word *farmhouse*.

the CEFR level and can be seen as complementary to the CEFR level and the frequency features, all of which focus on the surface lemma (frequency of all senses or homonyms, the lowest CEFR level, regardless of sense). We recognize words as lexically ambiguous if the test item refers to a less common sense or a homonym.

Word length is surprisingly one of the consistently relevant features. While length certainly contributes to the difficulty in acquiring a word, it also simply increases the probability of typos, even in cases when the writer knows the word’s spelling.

L1 calque is one of the globally least important features, reflecting both the frequency of calques in the language pairs and the fact that an L1 word being a calque (morpheme-for-morpheme translation) being less conducive to eliciting the production of the correct English word than the surface level L1 similarity.

Local explanations of all test set predictions of the explainable model can be viewed in an online

application.³ In Figure 2 we provide an example of the local explanation of the difficulty of the English word *farmhouse*, which, despite being relatively rare and not being assigned any CEFR level in the EVP, is not a particularly difficult word for L1 Chinese and German learners. Both its Chinese and German counterparts are calques of the English word *farmhouse*, which contributes to lower difficulty in these languages. On the other hand, for L1 Spanish speakers, the word is assessed as comparatively difficult to spell.

Neither the global nor the local explanations can be interpreted as the actual causes of the difficulty of a particular word. In particular, it cannot tell us anything about features not present in the model, and it underestimates the importance of a factor when the data has low quality (e.g., production frequency in the case of L1 German). On the other hand, given the relatively high accuracy of the model, we believe that the globally high-impact features provide valuable insights.

Both the high impact of production frequency and spelling difficulty is in line with the focus of the KVL. However, the combined importance of features reflecting the difficulty of spelling a word without errors (spelling difficulty, word length, and, to some extent, also the learners’ written production frequency) could affect the utility of the dataset for other uses, such as creating reading materials matched to the learner’s level. (Schmitt et al., 2024) suggest the KVL data could be more suitable for this purpose than frequency.

Given that the KVL are designed to focus on the most common word sense (Schmitt et al., 2024), the high importance of lexical ambiguity in our analysis could seem spurious. We have, however, observed uncommon word senses in the data. For instance, the noun *wireless* is included in the sense ‘radio (receiving set)’, which is dated and mainly British according to the Oxford Dictionary of English, rather than in the sense ‘wireless broadcasting or networking using radio signals’. For other words, such as *log* (‘chunk of wood’ in KVL, not ‘record’) or *diet* (‘nutrition’ in KVL, not ‘dietary regime’), two senses may be similarly common. Lexical ambiguity provides a useful signal in all these cases.

³<https://ynklab.github.io/vocabulary-difficulty/>

English word	Spanish word	Top LLM response
instantly	inmediatamente	immediately*
motivational	estimulante	motivating*
(to) reform	enmendar	revise
(to) amount	sumar	add up
everybody	todo el mundo	everyone*
synonymous	sinónimo	synonym*

Table 6: Examples of tricky test items in L1 Spanish, based on responses of GPT-4.1-mini. First letter printed in bold. Asterisks mark incorrect word length.

6.3 Tricky Test Items

The shared task rules permitted us to use the trickiness feature only in the open-track submission, where its impact was relatively small. However, it provides an interesting insight into how the test item construction often affects the resulting difficulty scores. The value of our trickiness feature is the probability of an LLM answering the test item incorrectly. Table 6 shows examples of high-trickiness items from the test set. For brevity, we focus on cases that do not depend on the L1 context. As exemplified by the items *instantly*, *motivational*, and *reform*, in many cases, the creators of the KVL test items intentionally avoided cognate L1 words. The words they have chosen instead often correspond better to English words other than the intended ones. In other cases, the reason is not the avoidance of cognates, but the L1 word strongly suggests a different English word nonetheless. While in some cases the L1 context or the number of letters disambiguates the English word (e.g., *sinónimo* is used as an adjective, not a noun; *everybody* and *everyone* differ in word length), such details were likely easy to miss not only for the LLM but also for the respondents. We believe that while better choices could have been made for some items, this shows an inherent limitation of the test item format used to compile the KVL.

7 Conclusion

We proposed a novel method for fine-tuning LLMs and MLMs to predict continuous values using cross-entropy loss with soft targets. The method achieved consistent gains on the vocabulary difficulty prediction task over prior approaches, such as discretization for LLMs and a regression head with MSE loss for MLMs.

In our fine-tuning experiments, increases in base model size up to 32B parameters resulted in performance improvements. The larger-sized LLMs

also consistently outperformed the smaller-sized MLMs. This result differs from the LCP study by Smádu et al. (2024), where much smaller fine-tuned MLMs performed comparably to much larger LLMs. We hypothesize that this could be due to the more complex nature of vocabulary difficulty prediction compared to LCP.

We built a separate explainable model that achieves competitive results on the task and provides insights into what affects vocabulary difficulty scores in the British Council’s KVL data. The high impact of spelling difficulty and the construction of the test items call for further investigation.

Limitations

We tested the proposed method for fine-tuning LLMs and MLMs to predict continuous values using cross-entropy loss with soft-targets only on this shared task’s data (KVL). Its suitability for other tasks and settings is yet to be investigated. On smaller or noisier data, for instance, its impact may be less pronounced. Various parameters, such as the number of points on the scale used, may also affect its performance.

The insights provided by our explainable model and the trickiness model can, in principle, be empirically validated, but we have not been able to do so, as the individual responses used to compile KVL are not publicly available. The frequency of spelling mistakes would be particularly easy to validate with access to the responses.

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.
- BNC Consortium. 2007. [British National Corpus, XML edition](#). <https://llds.ling-phil.ox.ac.uk/llds/xmlui/handle/20.500.14106/2554>.
- Annette Capel. 2012. [Completing the English Vocabulary Profile: C1 and C2 vocabulary](#). *English Profile Journal*, 3:e1.
- Tianqi Chen and Carlos Guestrin. 2016. [XGBoost: A Scalable Tree Boosting System](#). In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 785–794, New York, NY, USA. Association for Computing Machinery.

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- DeepSeek-AI. 2025. [DeepSeek-V3 Technical Report](#). *ArXiv preprint*, arXiv:2412.19437v2 [cs.CL].
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [QLoRA: Efficient Fine-tuning of Quantized LLMs](#). *Advances in Neural Information Processing Systems*, 36:10088–10115.
- Taisei Enomoto, Hwichan Kim, Toshio 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)*, pages 590–598, Mexico City, Mexico. Association for Computational Linguistics.
- Mariano Felice and Lucy Skidmore. 2026. Findings of the BEA 2026 shared task on vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. [Distilling the knowledge in a neural network](#). In *NIPS Deep Learning and Representation Learning Workshop*.
- 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.
- John H.A.L. Jong, Mike Mayor, and Catherine Hayes. 2016. [Developing global scale of English learning objectives aligned to the common European framework](#). Technical report.
- Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. [OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- 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.
- Scott M. Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, pages 4768–4777, Red Hook, NY, USA. Curran Associates Inc.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. [mmBERT: A Modern Multilingual Encoder with Annealed Language Learning](#). *ArXiv preprint*, arXiv:2509.06888v1 [cs].
- Mistral AI. 2026. [Ministral 3](#). *ArXiv preprint*, arXiv:2601.08584v1 [cs.CL].
- 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.
- Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2024. [Difficult for whom? a study of Japanese lexical complexity](#). In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 69–81, Miami, Florida, USA. Association for Computational Linguistics.
- Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2025a. [A Japanese Dataset and Efficient Multilingual LLM-Based Methods for Lexical Simplification and Lexical Complexity Prediction](#). *Journal of Natural Language Processing*, 32(4):1129–1188.
- Adam Nohejl, Frederikus Hudi, Eunike Andriani Kardinata, Shintaro Ozaki, Maria Angelica Riera Machin, Hongyu Sun, Justin Vasselli, and Taro Watanabe. 2025b. [Beyond film subtitles: Is YouTube the best approximation of spoken vocabulary?](#) In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9566–9585, Abu Dhabi, UAE. Association for Computational Linguistics.
- OpenAI. 2024. [GPT-4 Technical Report](#). *ArXiv preprint*, arXiv:2303.08774v6 [cs.CL].
- OpenAI. 2025. [Update to GPT-5 System Card: GPT-5.2](#). Technical report.
- 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.
- Qwen Team. 2025. [Qwen2.5 Technical Report](#). *ArXiv preprint*, arXiv:2412.15115v2 [cs.CL].

- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter](#). In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*, volume arXiv:1910.01108.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. [Introducing Knowledge-based Vocabulary Lists \(KVL\)](#). *TESOL Journal*, 12(4):e622.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. [Knowledge-Based Vocabulary Lists](#). British Council Monographs on Modern Language Testing. University of Toronto Press.
- 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(3):1258–1270.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Pérez Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 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 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. [Transformer architectures for vocabulary test item difficulty prediction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.
- Răzvan-Alexandru Smădu, David-Gabriel Ion, Dumitru-Clementin Cercel, Florin Pop, and Mihaela-Claudia Cercel. 2024. [Investigating large language models for complex word identification in multilingual and multidomain setups](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16764–16800, Miami, Florida, USA. Association for Computational Linguistics.
- Team GLM. 2024. [ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools](#). *ArXiv preprint*, arXiv:2406.12793v2 [cs.CL].
- 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 Prompt Templates

In the following, we list the full text of our prompt templates by feature for which they were used. The symbol ↵ stands for line breaks. Placeholders are printed in bold. The following common placeholders corresponding to the dataset items are used across multiple prompts:

- {l1_name}: an L1, i.e., Spanish, Chinese, or German.
- {l1_word}: an L1 word.
- {l1_context}: an L1 context.
- {en_word}: an English word.
- {clue}: a letter-pattern clue derived from the English word (first letter + space-separated underscores), e.g., b _ _ _ for book.

A.1 Fine-Tuning

Basic template:

Rate how difficult it is for learners to guess the English word based on the {l1_name} word, context and clue on a scale from 1 to 5 (1=very easy, 5=very difficult).↵
 {l1_name} word: {l1_word}↵
 {l1_name} context: {l1_context}↵
 Clue: {clue}↵
 English word: {en_word}↵
 Difficulty:

Short prompt template:

Used in the “short prompt” ablation.

```
{l1_word} ### {l1_context} ### {clue} ###
{en_word} ### Difficulty (1 to 5):
```

Regression template

Used for MLMs when fine-tuning with a regression head, following Skidmore et al. (2025).

```
[CLS] {l1_word} [SEP] {l1_context} [SEP] {clue}
[SEP] {en_word} [SEP]
```


A.2 Lexical Ambiguity

Placeholders:

We use the English word “bank” as an example of lexical ambiguity (‘financial institution’ vs. ‘(river) side’). For L1 Spanish, the placeholders would take the following values:

- {ex_en_word}: “bank”
- {ex_easy_word_l1}: “banco”
- {ex_easy_context_l1}: “Depositó el dinero en el banco.”
- {ex_hard_word_l1}: “orilla”
- {ex_hard_context_l1}: “Nos sentamos en la orilla del río.”

Prompt template:

```
You are a language education expert.↵
TASK↵
Given:↵
- an English word form (the "English word"),↵
- an L1 gloss/translation (the "{l1_name} item"),↵
- and the L1 usage context sentence (the "{l1_name} context"),↵
decide whether the English word, when used to
express the meaning suggested by the L1 item +
context,↵
meets BOTH conditions:↵
A) Lexical ambiguity: the English word has
multiple established senses that share the same
form↵
    (polysemy or homonymy), such that another
    common sense could plausibly be
    activated/confused.↵
B) Unfamiliarity for L2 learners: in this
meaning/usage, the English word is likely to be
unfamiliar↵
    or challenging for typical second-language
    learners (e.g., less frequent sense,
    idiomatic/figurative,↵
    domain-specific usage, nonliteral
    extension).↵
OUTPUT REQUIREMENTS↵
- Output "1" if BOTH conditions (A and B) are
met; otherwise output "0".↵
- Output MUST be exactly one character: 1 or 0.↵
- Do NOT include explanations, alternatives,
quotes, or extra text.↵
EXAMPLE 1↵
English word: {ex_en_word}↵
{l1_name} item: {ex_easy_word_l1}↵
{l1_name} context: {ex_easy_context_l1}↵
Is the English word ambiguous and unfamiliar: 0↵
EXAMPLE 2↵
English word: {ex_en_word}↵
{l1_name} item: {ex_hard_word_l1}↵
{l1_name} context: {ex_hard_context_l1}↵
Is the English word ambiguous and unfamiliar: 1↵
NOW DECIDE↵
English word: {en_word}↵
{l1_name} item: {l1_word}↵
```

```
{l1_name} context: {l1_context}↵
Is the English word ambiguous and unfamiliar:
```

A.3 Spelling Difficulty

This prompt is executed for all L1s at once.

Placeholders:

- {en_pron}: CMU-style pronunciation string of the English word of the current item. Example: K Y UW for “queue”.
- {all_l1_words[l1]}: L1 words of the current item (all three in one prompt).
- {hard_pron}, {hard_cn}, {hard_es}, {hard_de}, {hard_cn_score}, {hard_es_score}, {hard_de_score}: hard demonstration item (pronunciation, three L1 translations, and example scores 5, 4, 4).
- {easy_pron}, {easy_cn}, {easy_es}, {easy_de}, {easy_cn_score}, {easy_es_score}, {easy_de_score}: easy demonstration item (pronunciation, three L1 translations, and example scores 1, 1, 1).

Prompt template:

```
TASK↵
You are required to rate English spelling
difficulty on a 1-5 scale, where 1 = very easy
and 5 = very difficult.↵
You will be given English pronunciation and the
target word's translation in Chinese, Spanish,
and German.↵
Evaluate how difficult it would be for learners
with Chinese, Spanish, and German L1 backgrounds
to spell the English word with that pronunciation
correctly when they know the translation in their
native language.↵
OUTPUT REQUIREMENTS↵
- Output exactly one digit (1, 2, 3, 4, or 5) for
each L1, separated by commas, in the order of
Chinese, Spanish, German.↵
- Do not include any other text.↵
EXAMPLE 1↵
English pronunciation: '{hard_pron}'↵
Chinese: {hard_cn}↵
Spanish: {hard_es}↵
German: {hard_de}↵
Result:
{hard_cn_score},{hard_es_score},{hard_de_score}↵
EXAMPLE 2↵
English pronunciation: '{easy_pron}'↵
Chinese: {easy_cn}↵
Spanish: {easy_es}↵
German: {easy_de}↵
Result:
{easy_cn_score},{easy_es_score},{easy_de_score}↵
NOW DECIDE↵
English pronunciation: {en_pron}↵
Chinese: {all_l1_words['cn']}↵
Spanish: {all_l1_words['es']}↵
German: {all_l1_words['de']}↵
Result:
```

A.4 L1 Calque

This prompt uses Spanish examples for all L1s.

Prompt template:

You are a linguist and your task is to decide whether an English word is a morpheme-for-morpheme translation of any of the given **{l1_name}** equivalents.↵
The morpheme-for-morpheme mapping must be 1:1. 1:N or other mappings do not count.↵
Single morpheme translations or simple borrowings/cognates do not count either.↵
Respond only with YES or NO.↵
wave/ola: NO (reason: single morpheme)↵
ecosystem/ecosistema: NO (reason: simple cognate)↵
hotdog/perro caliente: YES (reason: hot=caliente, dog=perro)↵
stare/mirar fijamente: NO (reason: not a 1:1 mapping)↵
{en_word}/{l1_word}:

A.5 L1 Calque (used only in closed_max)

This was our first iteration of the prompt. It did not give the expected results (monomorphemic word pairs were labeled as calques), so we did not use it for our explainable model. However, as it performed well as a feature, we included it in the closed_max model.

Placeholders:

- **{ex_calque_l1}**: the L1-side word, e.g., “热狗” for Chinese, composed of the morphemes ‘hot’ and ‘dog’.
- **{ex_calque_en}**: the English-side word, e.g., “hot dog”.

Prompt template:

You are a bilingual linguistics expert.↵
TASK↵
Given a **{l1_name}** item and an English item, decide whether there exists a best-matching candidate in the **{l1_name}** item that is a component-by-component (morpheme-level) translation of the English item.↵
A component-by-component mapping means that the meaningful parts↵
(words, roots, prefixes, or suffixes) of the English item are directly translated↵
into corresponding meaningful parts in the **{l1_name}** item.↵
Procedure (internal; do NOT output these steps):↵
1) If the **{l1_name}** item contains multiple candidates, select exactly ONE candidate: the one that aligns best component-wise with the English form.↵
2) Judge ONLY that selected candidate for component-by-component mapping.↵
OUTPUT REQUIREMENTS↵
- Output “1” if the selected best candidate is a component-by-component mapping; otherwise output “0”.↵

- Output MUST be exactly one character: 1 or 0.↵
- Do NOT include explanations, alternatives, quotes, or extra text.↵
EXAMPLE↵
{l1_name} item: **{ex_calque_l1}**↵
English item: **{ex_calque_en}**↵
Is word-for-word mapping: 1↵
NOW DECIDE↵
{l1_name} item: **{l1_word}**↵
English item: **{en_word}**↵
Is word-for-word mapping:

A.6 Trickiness (short prompt, used only in open_max)

Placeholders:

{solve_example}: a formatted one-shot example (L1 word, L1 context, English word) using the item English word “strawberry”. Example:

German word: Erdbeere↵
German context: Ich mag keine Erdbeeren.↵
English word: strawberry

Prompt template:

You are bilingual in **{l1_name}** and English and your task is to find the best English translation for a **{l1_name}** word given a context and constraints. The constraints are given in the form of a clue, e.g., “b _ _”, meaning that the word starts with the (upper or lower case) letter B and has 4 letters. You must give a single English word in dictionary form (lemma) as a response.↵
{solve_example}↵
{l1_name} word: **{l1_word}**↵
{l1_name} context: **{l1_context}**↵
Clue: **{clue}**↵
English word:

A.7 Trickiness (long prompt, used only in open_max)

Placeholders:

Same as in [Section A.6](#).

Prompt template:

You are bilingual in **{l1_name}** and English.↵
TASK↵
Given a word in **{l1_name}**, its usage context, and a spelling clue, find the single best English translation that fits BOTH the meaning and the spelling constraint.↵
INPUTS↵
- **{l1_name}** word: a single word to translate↵
- **{l1_name}** context: a sentence showing how the word is used↵
- Clue: a pattern such as “b _ _”, where:↵
* the first letter is indicated
(case-insensitive)↵
* “_” indicates subsequent unknown letter↵
* the total number of letters must match exactly↵
OUTPUT REQUIREMENTS↵
- Output EXACTLY ONE English word↵
- The word must be:↵
* a dictionary form (lemma)↵

- * a single token (no spaces, hyphens, or punctuation)↵
- * consistent with the context↵
- * consistent with the clue↵
- Do NOT include explanations, alternatives, quotes, or extra text.↵

EXAMPLES↵

{solve_example}↵

NOW SOLVE↵

{l1_name} word: **{l1_word}**↵

{l1_name} context: **{l1_context}**↵

Clue: **{clue}**↵

English word:

A.8 Difficulty

Placeholders:

{examples}: a block of examples for difficulty rating using examples from the training data. Examples with ratings converted to values close to 1, 3, and 5 are selected for each L1 separately.

Prompt template:

You are an English language teacher teaching learners whose native language is **{l1_name}**. Your task is to rate the difficulty of a vocabulary test item for native **{l1_name}** speakers learning English.↵

The test item consists of:↵

- a **{l1_name}** word,↵
- a **{l1_name}** context,↵
- a clue indicating the first letter and word length of the English word,↵
- the target English word, which is the only correct answer.↵

Letter case does not matter. The learners are likely to respond with synonyms or misspellings to some items, but such responses are considered incorrect. Treat this as increasing the difficulty.↵

Consider learners from beginner to advanced levels, weighting the intermediate learner most heavily. Rate how difficult the item is on a scale from 1 to 5:↵

1 = very easy (almost everybody answers correctly)↵

5 = very difficult (almost nobody answers correctly)↵

Output exactly one digit (1, 2, 3, 4, or 5). Do not include any other text.↵

{examples}↵

{l1_name} word: **{l1_word}**↵

{l1_name} context: **{l1_context}**↵

Clue: **{clue}**↵

English word: **{en_word}**↵

Difficulty:

B Base Models and API Models

Table 7 lists base LLMs and MLMs that we fine-tuned in our experiments. Table 8 lists LLMs that we used for prompt-based features via API.

Model Name	Hugging Face Model ID	Systems/References to Results
GLM-4-32B	zai-org/GLM-4-32B-Base-0414	finetuned_llms*
Qwen2.5-32B	Qwen/Qwen2.5-32B	finetuned_llms*
Minstral-3-14B	mistralai/Minstral-3-14B-Base-2512	finetuned_llms*; $\leq 14B$ LLM Average; ablations in Table 4
Qwen2.5-14B	Qwen/Qwen2.5-14B	$\leq 14B$ LLM Average
GLM-4-9B	zai-org/glm-4-9b	$\leq 14B$ LLM Average; $\leq 9B$ LLM Average
Qwen2.5-7B	Qwen/Qwen2.5-7B	$\leq 9B$ LLM Average
Minstral-3-8B	mistralai/Minstral-3-8B-Base-2512	$\leq 9B$ LLM Average
Qwen2.5-1.5B	Qwen/Qwen2.5-1.5B	model size comparison in Appendix E
Qwen2.5-0.5B	Qwen/Qwen2.5-0.5B	model size comparison in Appendix E
mmBERT-b (mmBERT-base)	jhu-clsp/mmBERT-base	closed_max; regression and ablations in Table 3
XLMR-b (XLM-RoBERTa-base)	xlm-roberta-base	baselines; regression in Table 3
XLMR-l (XLM-RoBERTa-large)	xlm-roberta-large	regression in Table 3

*The three models used in finetuned_llms, were also used in the finetuned_llms_plus and open_max submissions, and the $\leq 32B$ LLM Average ensemble.

Table 7: Open-weight base models used in fine-tuning experiments.

Model Name	Provider	Model ID (Snapshot)	Explainable Features (explainable; open_max)	Additional Features (open_max)
GPT-5.2	OpenAI	gpt-5.2-2025-12-11	Lexical ambiguity; Spelling difficulty; L1 calque	Difficulty
GPT-4.1	OpenAI	gpt-4.1-2025-04-14		Trickiness; Difficulty
GPT-4.1-mini	OpenAI	gpt-4.1-mini-2025-04-14		Trickiness
GPT-4.1-nano	OpenAI	gpt-4.1-nano-2025-04-14		Trickiness
DeepSeek-V3.2	DeepSeek	deepseek-chat	Lexical ambiguity	

Table 8: Models used for features based on LLM prompting.

C Hyperparameters

Table 9 shows the general fine-tuning hyperparameters. For XLM-RoBERTa, we followed Skidmore et al. (2025). For other models, we performed a limited search for the learning rate and epochs using cross-validation. For mmBERT, a higher learning rate would likely be optimal, but $3e-5$ was the largest we evaluated. For LLMs, we found that the learning rate of $1e-4$ performed better than $1.5e-4$, but we could not rerun the training for Qwen2.5-32B in time for the final submission. Table 10 lists the quantization and LoRA hyperparameters used for all LLM fine-tuning.

For the XGBoost regressor (XGBRegressor in XGBoost’s Scikit-Learn API⁴), we set max_depth=3, learning_rate=0.1, and n_estimators=200, while using default values for other hyperparameters.

Model	Epochs	Batch size	Grad. accum.	Learning rate	Weight decay	Warmup ratio
XLM-RoBERTa (both)	5	32	—	$3e-5$ (linear)	0.1	0.1
mmBERT-base	16	16	—	$3e-5$ (const.)	0.1	0.1
Qwen2.5-32B	4	2	8	$1.5e-4$ (const.)	—	—
All other LLMs	4	2	8	$1e-4$ (const.)	—	—

Table 9: General fine-tuning hyperparameters used for different models.

⁴https://xgboost.readthedocs.io/en/release_3.1.0/python/python_api.html#xgboost.XGBRegressor

Group	Hyperparameter	Value
Quantization	Bit-width	4-bit
	Data type	NFloat4
	Double quant.	Yes
QLoRA-related	Compute type	BFloat16
	Optimizer	paged_adamw_8bit
LoRA adapter	Rank r	8
	Scaling α	16
	Dropout rate	0.05
	Target modules	All linear
	Bias adaptation	No

Table 10: Quantization and LoRA hyperparameters used for all LLM fine-tuning experiments.

D Results Reported as Pearson’s Correlation Coefficients

Table 11, Table 12, Table 13, and Table 14 show Pearson’s correlation coefficient (PCC) evaluation corresponding to the RMSE evaluation shown in the main text in Tables 2 to 5.

E Comparison of Models Across Sizes

In Table 15 and Table 16, we compare all base models we used in experiments, including models that are only reported in ensembles in the main text. To facilitate the comparison of models of different architectures (decoder LLMs and encoder MLMs), we also add Qwen models of sizes 1.5B and 0.5B, as well as XLM-RoBERTa models fine-tuned with the same prompt and method. We can observe that the model size, rather than the architecture, determines performance in this task. At the same size, however, MLMs are more compute-efficient.

All models were fine-tuned using the same cross-entropy loss with soft targets and the same prompt, utilizing hyperparameters listed in Appendix C.

F Simulation of a Statistical Optimum

The difficulty scores provided by the KVL data are based on test item responses. The number of responses collected for each item (120 to 228 responses) determines the precision of the data, which can be expressed as a confidence interval. Schmitt et al. (2024, Sec. 6.3, App 5) provide 83% confidence intervals for selected bands of the complete KVL data (e.g., items ranked 200–299 for L1=Spanish) expressed in numbers of ranks (e.g., 44). The rationale for 83% is that it corresponds to a significance level of 5% for differences in pairwise ordering (e.g., there is no significant difference

System	Chinese	German	Spanish	Mean
open_max	0.927	0.916	0.920	0.921
finetuned_llms_plus	0.928	0.915	0.919	0.921
finetuned_llms	0.925	0.914	0.915	0.918
≤32B LLM Average	0.926	0.914	0.916	0.918
- GLM-4-32B	0.918	0.907	0.906	0.910
- Qwen2.5-32B	0.917	0.902	0.907	0.909
- Ministral-3-14B	0.915	0.902	0.907	0.908
≤14B LLM Average	0.921	0.905	0.908	0.911
≤9B LLM Average	0.916	0.901	0.900	0.906
Open-Track Baseline	0.804	0.786	0.783	0.791
Statistical Optimum	0.989	0.990	0.995	0.991

Table 11: PCC of our open-track submissions, compared with average ensembles by model size, individual models, and the shared task’s open-track baseline.

System	Chinese	German	Spanish	Mean
closed_max	0.874	0.844	0.854	0.857
explainable	0.837	0.779	0.789	0.802
traditional	0.767	0.747	0.721	0.745
exp.: std. inference	0.820	0.768	0.776	0.788
exp.: lin. regression	0.815	0.766	0.769	0.783
Closed-Track Baseline	0.753	0.773	0.765	0.764

Table 12: PCC of our closed-track submissions, compared with two variants of the explainable model and the shared task’s closed-track baseline.

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (Ministral-3-14B)	0.915	0.902	0.907	0.908
- single language	0.909	0.890	0.907	0.902
- out-of-language	0.862	0.879	0.876	0.872
- short prompt	0.911	0.899	0.906	0.905
- standard loss	0.892	0.885	0.886	0.887
- std. loss & inference	0.859	0.853	0.850	0.854

Table 13: PCC of ablations of our LLM-based model.

in difficulty between items within ranks 200 and 44). The widest confidence intervals are 69, 95, and 108 ranks for Spanish, Chinese, and German, respectively. While the span of 100 ranks is given as a rule-of-thumb criterion for “strong confidence”, we use the above per-L1 maximum widths w for our simulation.

We simulate our statistical optimum predictions by taking the most distant difficulty in the complete KVL Data within $\pm w$ ranks of the predicted data point and using it as a prediction. In accordance with the originally reported confidence intervals, this process is based on the complete KVL data (training, development, and test subsets), although we predict only for the test data. Such predictions could be considered to have no statistically

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (mmBERT-b)	0.837	0.837	0.826	0.833
- single language	0.839	0.816	0.824	0.826
- out-of-language	1.115	1.102	1.193	1.136
- short prompt	0.834	0.827	0.825	0.828
- standard loss	0.807	0.794	0.788	0.796
- std. loss & inference	0.779	0.772	0.776	0.776
Regression (XLMR-b)	0.710	0.729	0.693	0.711
Regression (XLMR-l)	0.813	0.823	0.801	0.812
Reg. (mmBERT-b)	0.812	0.798	0.797	0.802

Table 14: PCC of ablations of our MLM-based model, compared with using a standard regression head and different base models (XLM-RoBERTa base/large).

Base Model	Chinese	German	Spanish	Mean
GLM-4-32B	0.678	0.769	0.805	0.751
Qwen2.5-32B	0.678	0.777	0.799	0.752
Ministral-3-14B	0.681	0.781	0.799	0.753
Qwen2.5-14B	0.701	0.801	0.832	0.778
GLM-4-9B	0.744	0.858	0.930	0.844
Ministral-3-8B	0.723	0.809	0.835	0.789
Qwen2.5-7B	0.722	0.842	0.882	0.816
Qwen2.5-1.5B	0.800	0.955	0.998	0.918
Qwen2.5-0.5B	0.851	1.021	1.078	0.983
XLMR-l (550B)	0.924	1.001	1.095	1.007
mmBERT-b (307M)	0.921	0.984	1.063	0.989
XLMR-b (270B)	1.042	1.136	1.220	1.133

Table 15: RMSE of all individual base LLMs and MLMs.

Base Model	Chinese	German	Spanish	Mean
GLM-4-32B	0.918	0.907	0.906	0.910
Qwen2.5-32B	0.917	0.902	0.907	0.909
Ministral-3-14B	0.915	0.902	0.907	0.908
Qwen2.5-14B	0.909	0.895	0.897	0.900
GLM-4-9B	0.900	0.880	0.874	0.885
Ministral-3-8B	0.904	0.895	0.898	0.899
Qwen2.5-7B	0.904	0.884	0.884	0.890
Qwen2.5-1.5B	0.883	0.848	0.848	0.860
Qwen2.5-0.5B	0.863	0.824	0.820	0.836
XLMR-l (550B)	0.835	0.832	0.816	0.827
mmBERT-b (307M)	0.837	0.837	0.826	0.833
XLMR-b (270B)	0.786	0.775	0.761	0.774

Table 16: PCC of all individual base LLMs and MLMs.

significant difference from the gold standard data; hence, we report their RMSE and correlation as a “statistical optimum”.

G Predicting in the Probability Space

As shown in Table 17, fine-tuning an LLM to predict in the probability space instead of the logit space, in which the GLMM scores are, resulted in a decrease in performance. We therefore used the raw GLMM scores for the main experiments.

Target Values	↓ RMSE				↑ PCC			
	Chinese	German	Spanish	Mean	Chinese	German	Spanish	Mean
Logits: GLMM score	0.681	0.781	0.799	0.753	0.915	0.902	0.907	0.908
Probabilities: expit(GLMM score)	0.738	0.853	0.877	0.822	0.905	0.889	0.895	0.896

Table 17: Performance of Ministral-3-14B when fine-tuned on the raw GLMM score (logits), as we did in all experiments, compared with the same model fine-tuned on the corresponding probabilities. In both cases, the values were linearly transformed to the 1-to-5 difficulty scale we use in our prompt, and all other parameters were the same. RMSE and PCC were measured for GLMM score values in the logit space for both.

Report on the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German

Sebastian Gombert¹, Zhifan Sun¹, Fabian Zehner^{1,2}, Jannik Lossjew³, Tobias Wyrwich⁴,
Berit K. Czinczel⁴, David Bednorz⁴, Sascha Bernholt⁴, Knut Neumann⁴,
Ute Harms⁴, Aiso Heinze⁴, Hendrik Drachsler^{1,5,6}

¹DIPF | Leibniz-Institute for Research and Information in Education

²Centre for International Student Assessment (ZIB)

³Institute for Chemistry Education, Eberhard Karls University of Tübingen

⁴IPN | Leibniz Institute for Science and Mathematics Education

⁵Studiumdigitale & ⁶Computer Science Department, Goethe University Frankfurt

Abstract

We present the BEA 2026 shared task on rubric-based short answer scoring for German. Rubric-based short answer scoring is a case of automatic short answer scoring (ASAS) that requires models to apply textual scoring rubrics to student answers as a basis for assigning scores. For the shared task, we introduced a novel German-language dataset from multiple STEM domains to provide a comprehensive benchmark for this problem. The dataset was designed to evaluate both performance and generalization (the latter, by distinguishing between seen and unseen questions), as well as coarse- and fine-grained scoring (2-way vs. 3-way). The systems submitted to the shared task cover a wide range of approaches, including fine-tuned large language models, prompt-based methods, human-AI collaboration strategies, or a combination of these. The results show that structured, task-adapted LLM systems achieved the strongest performance across all tracks. The winning system, IWM-DKM, combined LoRA fine-tuning of Qwen models with rubric-aware input structuring, including checklist-style reasoning, rubric reframing as decision trees, background knowledge injection, and ensemble voting. Other systems similarly relied on fine-tuned LLMs, retrieval-augmented prompting, encoder-LLM ensembles, or weighted aggregation strategies. Overall, the shared task results show that rubric-based scoring benefits most from systems that explicitly operationalise rubric semantics, while generalisation to unseen questions remains a central challenge.

1 Introduction

Short answer scoring is an established task in educational natural language processing (Burrows et al., 2015; Bai and Stede, 2023; Bexte et al., 2024). It was already addressed in at least two previous shared tasks, namely the *SemEval 2013 Shared Task* (Dzikovska et al., 2013) and the ASAP-

SAS Kaggle competition (Hamner et al., 2012). In recent years, it has primarily been approached using two paradigms: instance-based and similarity-based approaches (Bexte et al., 2022, 2023).

On the one hand, instance-based approaches train a model that predicts scores as labels or regression targets for a given input response. While this approach has been shown to work quite well, it also suffers from the problem of not transferring well to questions unseen during training (Padó et al., 2023). This means that implementing an instance-based short-answer scoring system requires collecting and manually scoring student answers for a select set of questions. A model trained on these in an instance-based fashion then only supports scoring unseen answers to these known questions. In this approach, the resulting model is intended to implicitly take up the relevant rules for scoring instances.

Similarity-based approaches, on the other hand, attempt to circumvent these problems by selecting a setup based on nearest-neighbour search or natural language inference (Bexte et al., 2022; Camus and Filighera, 2020), which are designed to compare a student’s answer to one or more provided reference answers. While this results in better transfer capabilities to unseen questions and domains, a problem with this setup is that not all possible question formats necessarily allow using a similarity-based setup, since for some questions, a broad range of answers might be valid, and out-of-sample generalization may fail due to unexpected content during inference.

With this shared task, we therefore look at another interpretation of the task that could help to circumvent these problems: rubric-based short answer scoring. For rubric-based short answer scoring, a model is provided with an answer, a question/prompt and a textual scoring rubric that, for each possible score, establishes criteria that need to be met so the same can be assigned to a given answer. To successfully solve this task, a model must

account for the semantics of a scoring rubric and apply its criteria to the candidate answers. This setup is inspired by established human scoring practice (Reddy and Andrade, 2010; Panadero and Jonsson, 2013). Here, human scorers are often provided with a rubric and then need to reason about which criteria apply to a given answer to assign the appropriate score. These rubrics are often question-specific and are, in turn, typically embedded in assessment frameworks that specify the construct that the test is intended to measure (e.g. OECD, 2024, 2023).

While rubrics have been applied in various works on free-text scoring and have shown promise as supplementary input or pre-training resource for instance- and similarity-based scoring systems (Wang et al., 2019; Li et al., 2023; Condor et al., 2022; Sonkar et al., 2024) as well as for systems based on LLM prompting (Wei et al., 2025; Frohn et al., 2025; Cong et al., 2026), there is, as of now, a lack of focused research on rubric-based short-answer scoring. However, this involves many challenges such as the uncertainty in the rubrics and, depending on the question, the semantic variability of the answers. With additional information from rubrics or assessment frameworks, resulting models can be designed more explicitly to incorporate scoring criteria that are specific to a given domain or assessment construct. As of now, there is a clear research gap in understanding how well current methods in NLP can comprehend rubrics and apply them to students’ answers to predict corresponding scores. With this shared task, we aimed to kickstart research on this specific interpretation of the short-answer scoring task with the following concrete contributions:

- We hosted a shared task on *rubric-based short answer scoring*, a setting that reflects real-world educational assessment, ranging from classroom to standardized testing, and requires models to interpret and apply textual scoring criteria.
- We provided a new publicly available benchmark dataset for this task in German, contributing to the currently limited pool of non-English resources for ASAS.
- We designed a comprehensive evaluation setup that systematically distinguishes between seen and unseen questions as well as coarse- and fine-grained scoring (2-way vs.

3-way), enabling a detailed analysis of generalization capabilities.

- In this paper, we present a comparative overview of the submitted modeling approaches, highlighting emerging design patterns such as structured LLM reasoning, prompt optimization, hybrid modeling, and human–AI collaboration.

2 Related Work

Automatic short answer scoring (ASAS) is an established task in educational NLP, where systems aim to assign scores to short written student responses (Burrows et al., 2015; Bai and Stede, 2023; Bexte et al., 2024). Recent work has mainly followed two underlying paradigms (Bexte et al., 2023). Instance-based approaches treat scoring as text classification or regression, where student answers are mapped directly to score labels (Riordan et al., 2017; Gombert et al., 2023). These methods work well for questions seen during training, but typically transfer less reliably to unseen questions (Padó et al., 2023), especially when the new questions assume a different number of levels. Similarity-based approaches instead compare student answers to reference answers or representative answer clusters, for example, using natural language inference or sentence embeddings (Bexte et al., 2022, 2023; Zehner et al., 2016). While this can improve reuse across questions, such methods depend on the availability of reference answers at all available levels and full semantic coverage is not always guaranteed.

Rubrics are central to human scoring practice because they define explicit criteria for assigning performance levels (Reddy and Andrade, 2010; Panadero and Jonsson, 2013). In automatic scoring, however, they have mostly been used as additional input rather than as the main scoring reference. Earlier work has injected rubric information into neural scoring models (Wang et al., 2019) or used rubrics for further pre-training (Condor et al., 2022). These studies suggest that rubrics can improve scoring, especially in low-resource and transfer settings, but focused research on rubric-based short answer scoring remains limited.

Recent work has started modelling rubrics as central semantic structures for scoring. Gombert et al. (2026b) introduced the *GRAASP* and *ToLe-GRAA* architectures, in which scoring is modelled as the alignment between student responses and

rubrics, using attention over transformer-based representations. Building on this direction, the *AGRAA* framework (Gombert et al., 2026a) models rubric descriptors as low-dimensional semantic subspaces derived from contextualised embeddings, so that scoring corresponds to measuring how strongly a response aligns with rubric-induced aspect spaces.

In parallel to model-based approaches, recent work has explored *rubric-based prompting* of large language models (LLMs) for automated assessment. Instead of fine-tuning, these approaches provide rubric criteria directly within the prompt and ask the model to assign scores based on them. Cong et al. (2026) showed that incorporating rubric descriptors into prompts substantially improves scoring performance compared to prompting without rubrics, highlighting that rubrics provide more structured guidance than standard few-shot examples. Similarly, Wei et al. (2025) demonstrate that *concept-based rubrics* not only improve LLM scoring performance but also enable the generation of synthetic training data for downstream supervised models. Overall, these findings suggest that even without task-specific training, LLMs can leverage rubric structures when they are explicitly encoded in prompts, although their performance and reliability remain sensitive to rubric design and task complexity.

3 Dataset: ALICE-LP 1.1

The dataset we used in this shared task is a novel dataset called *ALICE-LP 1.1*, which contains mid- and high-school-level answers to questions from four STEM domains: *physics* (Wyrwich et al., 2025), *mathematics* (Bednorz et al., 2024), *biology* (Czinczel et al., 2025), and *chemistry* (Lossjew and Bernholt, 2025). These answers were collected within the context of the *ALICE* project, funded by the Leibniz Foundation, at middle and high schools in the German state of Schleswig-Holstein. The main goal of this project was to conduct learning progression analytics (Kubsch et al., 2022) in the classroom. Teachers guided students through lessons which involved solving various interactive learning activities in Moodle courses synchronously. This included a set of short-answer items, which are the basis of this dataset.

ALICE-LP 1.1 is a German ASAS dataset. Each question is rated on a three-point scale (incorrect, partially correct, correct). Each datapoint consists of the question text, the answer, a textual two- as

Set	#Questions	#Answers	#Levels (0/1/2)
Train	78	7,899	2,728/2,863/2,308
UA	78	2,008	691/726/591
UQ	39	3,086	1,146/1,021/919

Table 1: Distributional properties of *ALICE-LP 1.1*.

well as three-way rubric defining the criteria for each possible score, and the corresponding score. Table 2 shows an exemplary datapoint from this dataset. The two-way rubric has been artificially created by merging the partially correct and incorrect score levels.

The answers were scored by multiple student assistants enrolled in teacher education programmes for the respective subjects. For each question, an agreement was measured using a subset of answers. Student assistants were trained to score each question using a subset of answers until an agreement of at least $\kappa \geq .65$ was reached. The remaining answers were then distributed among the raters. The training set includes a total of 7,899 answers. The evaluation set is divided into two subsets: *unseen answers* and *unseen questions*. This setup was inspired by the established SemEval-2013 (Dzikovska et al., 2013) and SAF (Filighera et al., 2022) datasets, enabling evaluation of how well systems transfer to unseen questions. The *unseen answers* dataset includes 2,008 answers to questions contained in the training set (using a question-wise stratified 80/20 split). In contrast, the *unseen questions* dataset contains 3,086 answers to questions not in the training set. An earlier version of this dataset (*ALICE-LP 1.0*) was used in Gombert et al. (2026b) and Gombert et al. (2026a). For the shared task and the published benchmark dataset, we removed one question with 82 responses from the unseen questions dataset because we found it requires a reference answer to be properly gradable. Table 1 shows the distributional properties of *ALICE-LP 1.1*.

4 Shared Task Structure

In the context of the shared task, we hosted four evaluation tracks. All tracks were evaluated using the same set of evaluation metrics: weighted precision, weighted recall, weighted F1, and quadratic weighted kappa (QWK). QWK served as the primary metric for leaderboard ranking for each track.

Unseen Answers 2-Way For this track, participating models were evaluated based on how well

Input Category	Example
Question	Name consequences that the gas shortage could have for Germany and your school.
Answer	The school might have to close because it can no longer be heated. It could also be that students just have to wear jackets during lessons.
Rubric	(2) Students identify at least two links between a gas supply stop and the supply of electricity and/or heating. (1) Students identify one link between a gas supply stop and the supply of electricity and/or heating. (0) Students do not identify a link between a gas supply stop and the supply of electricity and/or heating.
Score	0/1 (2-Way) 1/2 (3-Way)

Table 2: An example question with the corresponding rubric, one example student answer and the corresponding score taken from the *ALICE-LP* dataset (translated from German to English).

they could score unseen answers to questions seen during training. For this track, we distinguished only between correct and incorrect answers, with partially correct answers counted as incorrect.

Unseen Questions 2-Way For this track, participating models were evaluated based on how well they could score answers to questions not seen during training. For this track, we distinguished only between correct and incorrect answers, with partially correct answers counted as incorrect.

Unseen Answers 3-Way For this track, participating models were evaluated based on how well they could score unseen answers to questions seen during training. For this track, we distinguished between correct, partially correct, and incorrect answers.

Unseen Questions 3-Way For this track, participating models were evaluated based on how well they could score answers to questions not seen during training. For this track, we distinguished between correct, partially correct, and incorrect answers.

All four tracks were hosted on Codabench (Xu et al., 2022). Participants were allowed to submit up to ten submissions per track, with the best submission, as determined by QWK, counted towards the official leaderboard of the shared task. We allowed multiple submissions to account for different variants of authors’ systems and the non-deterministic nature of deep learning methods. In the Appendix, we provide tables of the five best runs per team and track.

5 Submissions

Except for the teams *Diffuser* and *Afrilan*, all participating teams submitted system description papers.

We summarise the results in the following section. All teams participated in the *Unseen Answers 2-way* track, making it the most complete comparison setting. For the *Unseen Questions 2-way* track, all teams except *Diffuser* submitted results. In the *Unseen Answers 3-way* track, again all teams except *HFT* participated, while the *Unseen Questions 3-way* track saw the lowest participation, with only *IWM-DKM*, *WSE Research*, *SDPA*, *ASLAN*, *Afrilan*, and *AMATI* submitting results. Overall, participation decreases slightly in more challenging and fine-grained settings, particularly in the unseen-question and 3-way-classification scenarios.

Most participating teams were affiliated with German institutions, which is expected, given that the dataset and task are centred on German-language student answers. At the same time, the shared task attracted participants from outside Germany, indicating broader international interest in rubric-based short-answer scoring. This suggests that, despite the language-specific setting, the underlying problem formulation is relevant beyond the German research community.

5.1 System Descriptions

While this section systematises the submissions’ methodology, Table 3 provides a brief summary.

IWM-DKM (Belcher et al., 2026) The approach is based on supervised fine-tuning of decoder-only LLMs (Qwen family) with LoRA, where the model jointly processes the question, student answer, and rubric to predict scores. Performance is improved by systematically enriching and structuring the input: (i) checklist thinking introduces intermediate yes/no decisions aligned with rubric levels to guide reasoning, (ii) rubric reframing converts rubrics into structured decision trees for clearer decision

Team	Keywords of the Approach	Models / LLMs (Final System)
IWM-DKM (Belcher et al., 2026)	LoRA fine-tuning + (checklist thinking, rubric reframing, background knowledge injection); ensemble voting	Qwen-3-4B (QLoRA), Qwen-3.5-9B (QLoRA)
WSE Research (Gwozdz and Both, 2026)	LoRA fine-tuning + (instruction-following format, 7B–72B scaling); rubric-conditioned prompting + (TF-IDF exemplar retrieval); ensemble aggregation + (QWK-weighted voting)	Qwen-2.5-72B (QLoRA), Gemini 3 Flash + (Prompting + tf-idf-based RAG), kNN with tf-idf
SDPA (Liu and Zhang, 2026)	LoRA/DoRA fine-tuning + (Alpaca instruction format, direct label prediction); ensemble + (trained encoder, few-shot Claude, optimised soft voting)	GELECTRA-large, XLM-RoBERTa-large, BERT-base-german-cased, Claude (few-shot), Llama 3.1 8B (LoRA/DoRA)
ASLAN (Bexte et al., 2026)	Similarity-based scoring + (LaSiLearn, SetFit); instance-based classification + (domain-specific fine-tuning); zero-shot rubric-based prompting; majority voting	G-SciEdBERT, LaSiLearn / SetFit + (mpnet, mini-lm, distil), Phi-4, Gemma 3, GPT-5-mini + (ensemble via majority voting)
AMATI (Willis and Third, 2026)	Inductive Logic Programming + (ASP-based rule induction, lemma matching, key information rules); rule-guided prompting + (ILP theory injection, in-context training examples)	Mistral combined with ILP-derived rules
RETUYT-INCO (Sastre et al., 2026)	Meta-prompting + (group-specific prompt generation, per-group selection); prompt tuning + (per-group soft prompt embeddings); synthetic data augmentation + (class balancing, noise injection)	Gemini 3 Flash
HFT (Padó, 2026)	Zero-shot rubric-based prompting + (role prompting, expert committee prompting); hybrid human–AI grading + (confidence-based routing, pre-defined quality threshold)	GPT-OSS-120B

Table 3: Overview of final systems submitted by participating teams, focusing on core modelling approaches and models used for official submissions.

boundaries, and (iii) background knowledge injection provides generated domain context to support understanding. These techniques are complemented by strategies such as skill-based data partitioning and tailored validation setups, and are ultimately combined in ensemble models using majority voting to increase robustness, particularly for ambiguous cases, with gains arising from the interaction of these components rather than any single method alone.

WSE Research (Gwozdz and Both, 2026) The approach combines three complementary scoring strategies within a unified pipeline: rubric-based prompting, LoRA-based fine-tuning of Qwen models, and prediction aggregation. Given the input triple (question, student answer, rubric), candidate scores are generated via prompted LLMs (optionally enhanced with TF-IDF-based example retrieval), multiple fine-tuned models across different scales, and a non-neural TF-IDF kNN baseline. Fine-tuning is performed in an instruction-following format that mirrors inference, enabling models to internalise rubric criteria more effec-

tively. Scaling experiments show that performance improves with model size but saturates at larger scales. Predictions are combined using weighted voting or stacking, where aggregation improves performance on seen questions by leveraging complementary error patterns, whereas for unseen questions, a single large fine-tuned model generalises best.

SDPA (Liu and Zhang, 2026) The approach employs parameter-efficient fine-tuning of LLMs for rubric-based short answer scoring under low-resource conditions, alongside an ensemble baseline. It compares two complementary paradigms: (i) an encoder-based ensemble that combines multiple fine-tuned transformer models with few-shot LLM predictions (e.g., Claude) via weighted soft voting, and (ii) instruction-tuned LLMs (Llama3.1-8B) adapted using PEFT methods such as LoRA and DoRA. The latter directly predicts rubric-aligned labels from structured inputs (question, answer, rubric), enabling the model to internalise rubric reasoning rather than relying on prompt-based inference. Results show that while ensem-

bles benefit from combining complementary signals, particularly on seen questions, parameter-efficient fine-tuned LLMs consistently outperform both prompt-based and ensemble approaches, especially for unseen questions where generalisation to new rubrics is critical.

ASLAN (Bexte et al., 2026) The approach combines three complementary scoring paradigms: similarity-based methods, instance-based classification, and rubric-prompted LLMs, and evaluates them across seen and unseen settings. Similarity-based models (e.g., SBERT variants with LaSiLearn or SetFit) compare student answers to labelled training answers, while instance-based models (e.g., BERT variants) learn direct mappings from input text to scores. For unseen questions, where no task-specific training data is available, zero-shot LLMs are prompted with the rubric to assign labels. Across all paradigms, predictions are combined using majority voting, leveraging their complementary strengths. Results show that similarity-based approaches perform best when training data for a question is available, while LLM-based methods are more suitable for generalising to unseen questions. Ensemble methods consistently improve robustness and overall performance by integrating signals from different scoring strategies.

AMATI (Willis and Third, 2026) The approach combines symbolic rule learning with large language models to balance interpretability and robustness. First, Inductive Logic Programming (ILP) is used to induce human-readable grading rules from training data, where questions, rubrics, and student answers are represented as logical predicates and optimised via Answer Set Programming to generate rule-based scoring theories. These rules capture lexical and structural patterns aligned with rubric criteria, enabling transparent and explainable predictions. To address the brittleness of symbolic rules, the system integrates them with an LLM (Mistral) by incorporating either training examples or induced rules into the prompt. While the LLM alone provides robust scoring, especially for noisy student language, combining it with induced rules improves performance in the more fine-grained 3-way setting.

RETUYT-INCO (Sastre et al., 2026) The approach centres on meta-prompting, where an LLM is used to automatically generate task-specific scoring prompts tailored to each question–rubric pair.

In an offline phase, the model analyses the rubric and labelled training answers for a given group to produce a reusable, optimised prompt that captures the scoring criteria more precisely than a generic template. At inference time, new student answers are routed to their corresponding group and evaluated using the generated prompt, effectively adapting the scoring behaviour to the specific characteristics of each task. To further improve performance, the method incorporates prompt variation, selection based on validation performance, and synthetic data generation to address class imbalance.

HFT (Padó, 2026) The approach explores a low-cost, blunt-edge grading setup based on zero-shot prompting of open-source LLMs, combined with a hybrid human–machine workflow. The system uses simple rubric-grounded prompts with off-the-shelf LLMs to produce initial grade predictions, prioritising robustness, reproducibility, and applicability in resource-constrained settings. While these models perform moderately well, especially in information-poor scenarios, they are not entirely competitive with fine-tuning-based approaches when richer task-specific data is available. To compensate for this, the method incorporates human review in a structured hybrid grading process: model predictions for the most reliable class are accepted automatically, while less certain cases are manually reviewed.

6 Results

Table 4 shows the best submitted result per team and track. Overall, the results reflect the methodological differences discussed above and reveal a consistent hierarchy across settings for the shared task’s dataset, *ALICE-LP 1.1*. In the information-specific Unseen Answers conditions that allow training question-specific models, strongly optimised, model-centric approaches dominate: *IWM-DKM* and *WSE Research* consistently achieve the top ranks across both 2-way and 3-way tasks, with *SDPA* closely following. This confirms that structured fine-tuning, input engineering, and ensembling are highly effective when scored answers are available for training. In contrast, more prompt-centric approaches (sensu instruction learning) such as *RETUYT-INCO* and multi-paradigm ensembles such as *ASLAN* achieve competitive but slightly lower performance, indicating that while prompt optimisation and multi-paradigm aggregation are beneficial, they do not fully match the

Rank	Team Name	↑ Quadratic Weighted Kappa	Weighted Precision	Weighted Recall	Weighted F1-Score
Unseen Answers 2-way					
1	IWM-DKM	.726	.887	.887	.887
2	WSE Research	.717	.882	.883	.883
3	SDPA	.682	.869	.866	.867
4	RETUYT-INCO	.674	.865	.861	.863
5	ASLAN	.663	.861	.864	.862
6	AMATI	.644	.857	.843	.847
7	Diffuser	.615	.840	.842	.841
8	Afrilan	.546	.814	.803	.807
9	HFT	.477	.786	.793	.787
Unseen Questions 2-way					
1	IWM-DKM	.550	.813	.818	.815
2	SDPA	.535	.806	.804	.805
3	WSE Research	.533	.806	.811	.808
4	RETUYT-INCO	.490	.787	.791	.789
5	HFT	.482	.786	.793	.788
6	ASLAN	.457	.780	.789	.779
7	Afrilan	.437	.766	.757	.761
8	AMATI	.289	.704	.714	.708
Unseen Answers 3-way					
1	IWM-DKM	.796	.781	.782	.780
2	WSE Research	.790	.773	.774	.773
3	SDPA	.776	.775	.763	.766
4	ASLAN	.757	.758	.743	.746
5	AMATI	.749	.760	.736	.739
6	RETUYT-INCO	.729	.728	.728	.728
7	Diffuser	.698	.712	.703	.705
8	Afrilan	.647	.659	.661	.658
Unseen Questions 3-way					
1	IWM-DKM	.681	.680	.664	.669
2	WSE Research	.672	.670	.663	.665
3	SDPA	.644	.634	.633	.633
4	ASLAN	.579	.653	.587	.593
5	Afrilan	.523	.607	.608	.607
6	AMATI	.394	.525	.525	.520

Table 4: The best result achieved by each team across the four evaluation tracks.

performance of fine-tuned decoder-based LLMs in these settings. The sequential combination of symbolic formulations used to prompt an LLM, as employed by *AMATI*, further trails these approaches.

In the Unseen Questions setting, however, the ranking partially differed. While *IWM-DKM* remained the strongest, the advantage of fine-tuning diminished, and approaches emphasising cross-question generalisation became more competitive: *SDPA* and *WSE Research* remained strong, but *RETUYT-INCO* and especially *HFT* closed the gap despite relying on prompt-based strategies. This highlights that without question-specific training signals, simpler, more generic approaches can likely perform comparably to complex systems. Notably, *AMATI* showed a pronounced drop in this setting. Across all tracks, *IWM-DKM*’s consistent first-place performance suggests that combining

structured reasoning, task-specific fine-tuning, and ensembling provided the most robust overall solution.

7 Synthesis

Across submissions, a clear structure emerges that can be interpreted along three main axes, with individual teams occupying different positions in this space.

First, several teams converged on model-centric, LLM-driven approaches that are heavily structured and optimised. This is most evident for *IWM-DKM*, *WSE Research*, and *SDPA*. All three relied on fine-tuned LLMs as the core component, but differed in how much additional structure they imposed: *IWM-DKM* explicitly injected reasoning structure via checklist thinking and rubric reframing, *WSE Research* focused on combining multiple model

variants and retrieval signals in a unified pipeline, and *SDPA* emphasised parameter-efficient adaptation and showed that carefully fine-tuned models can outperform more complex ensembles, particularly in generalisation scenarios. Despite these differences, the shared insight is that strong performance emerged when the LLM was not used “as-is”, but is systematically aligned with rubric structure, either through input design, fine-tuning, or both.

Second, a number of teams moved towards hybrid pipelines that combine complementary paradigms. This is most clearly represented by *ASLAN* and, to some extent, *WSE Research* and *SDPA*. Here, the task was decomposed into multiple views, e.g. similarity-based retrieval, classical classification, or LLM-based scoring, which are then aggregated via voting or stacking. *ASLAN* is the most explicit example of this philosophy, showing that different paradigms dominate under different conditions (e.g., similarity-based methods for seen questions vs. LLMs for unseen ones). This reinforces a broader pattern also visible in *WSE Research*: performance gains were often due to complementarity between models, rather than improvements within a single model class.

Third, several approaches focused less on the model itself and more on how the task is interfaced with or augmented around the model. *RETUYT-INCO* is the clearest example here, shifting the optimisation problem from model parameters to prompt generation, effectively learning task-specific scoring functions via meta-prompting. *AMATI* explored a different direction by injecting symbolic structure into the LLM via ILP-derived rules, trading some flexibility for interpretability and controllability. Finally, *HFT* represents the extreme end of this axis, deliberately using minimal, zero-shot LLM setups and compensating through a hybrid human–AI process in which model predictions were selectively reviewed.

8 Discussion

The results of the shared task highlight both the potential and the current limitations of rubric-based short-answer scoring. Across all tracks, approaches that explicitly align model behaviour with rubric structure achieved the strongest performance. In particular, fine-tuned systems with structured input representations consistently outperformed prompting-based approaches when sufficient task-

specific information was available. However, results on the unseen question tracks showed that this advantage did not fully transfer to more challenging generalisation scenarios. While top-performing systems remained competitive, the performance gap to simpler methods narrowed. In this context, prompt-based and zero-shot methods could provide a competitive alternative despite lower peak performance. Another key observation is the importance of hybrid approaches. Many submissions combined multiple components via ensembling, integrating different scoring paradigms, or incorporating symbolic elements. This suggests that rubric-based scoring likely benefits most from combining complementary signals rather than relying on a single modelling paradigm.

9 Conclusion

Overall, the results show that structured and task-adapted LLM systems currently achieve the strongest performance, especially when questions are seen during training. At the same time, the unseen question tracks demonstrate that generalisation is possible to a certain degree, but remains challenging. We hope that this shared task provides a foundation for future work on rubric-based scoring, especially with respect to better generalisation, more interpretable scoring processes, and practically useful systems for educational settings.

Acknowledgements

This work is based on data from the *ALICE* project, funded by the *Leibniz Association* (K365/2020).

Limitations

Several limitations of the shared task should be considered when interpreting the results. First, the datasets used in this task are limited in scope with respect to domain, rubric complexity and score granularity. The dataset was collected in the German K12 context from grades 7-11 in the state of Schleswig-Holstein. All items use relatively simple holistic rubrics with three performance levels, which restricts conclusions about how well current approaches scale to more complex, analytic rubrics commonly used in real-world assessment.

Second, while the inclusion of unseen question tracks allows us to study generalisation, the evaluation is still confined to a single dataset and domain distribution. As a result, it remains unclear to what extent the observed findings transfer to

other subjects, educational levels, or languages beyond the German setting considered here. Third, the shared task focuses primarily on predictive performance and does not systematically evaluate the interpretability or faithfulness of scoring decisions. Although some submitted systems explicitly aim to improve transparency, the current evaluation setup does not allow for a rigorous comparison of how well models align their decisions with rubric criteria.

Several limitations of the shared task arise from constraints in dataset construction and release. First, in some cases, the provided rubrics are underspecified for the full range of valid student answers. While this reflects authentic assessment practice to some degree, it can render the scoring task inherently ambiguous and may disadvantage models that rely on precise alignment between rubric criteria and responses. Second, for a subset of questions, relevant context materials could not be included due to legal restrictions. As a result, models were required to score answers without access to information that human raters would typically consider, which likely limited achievable performance and may have introduced additional variance across items.

Finally, while the shared task setup aims to approximate realistic assessment conditions, the combination of partial rubric specification and missing context materials means that the benchmark does not fully capture ideal scoring conditions for all questions.

Ethics Statement

With this shared task, we aimed to explore the potential of rubric-based scoring for short answers. Since short-answer scoring directly concerns the rating of students' performance, the ethicality of the technology depends heavily on the specific setting in which it is deployed. Technology for rating students' performance should match or outperform human performance when deployed for high-stakes assessments and be subject to human supervision. The EU AI Act labels AI in education as overall high stakes. However, we argue that there are also low-stakes scenarios in which rubric-based short answer scoring technologies can be deployed. These include typical low-stakes learning scenarios as provided by intelligent tutoring systems or comparable applications. Overall, significant downstream research on technology for

rubric-based short-answer scoring is likely needed before the respective technology can be deployed in practical scenarios.

References

- Xiaoyu Bai and Manfred Stede. 2023. A survey of current machine learning approaches to student free-text evaluation for intelligent tutoring. *International Journal of Artificial Intelligence in Education*, 33(4):992–1030.
- David Bednorz, Kristin Litteck, Daniel Sommerhoff, and Aiso Heinze. 2024. *Erfassung individueller Lerntrajektorien in einer digitalen Lernumgebung zum Ableitungsbegriff*. Universitätsbibliothek Dortmund.
- Kate Rebecca Belcher, Marius De Kuthy Meurers, Kordula De Kuthy, and Detmar Meurers. 2026. IWM-DKM at BEA 2026 shared task 2: Supplementing supervised fine-tuning for rubric-based short answer scoring. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Marie Bexte, Yuning Ding, Josef Ruppenhofer, Nils-Jonathan Schaller, Daniel Mora Melanchthon, Torsten Zesch, and Andrea Horbach. 2026. ASLAN at BEA 2026 shared task 2: Voting across scoring paradigms. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- 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.
- 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, Toronto, Canada. Association for Computational Linguistics.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2024. Strengths and weaknesses of automated scoring of free-text student answers. *Informatik Spektrum*, pages 1–9.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International journal of artificial intelligence in education*, 25:60–117.
- Leon Camus and Anna Filighera. 2020. Investigating transformers for automatic short answer grading. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 43–48. Springer.

- Aubrey Condor, Zachary Pardos, and Marcia Linn. 2022. Representing scoring rubrics as graphs for automatic short answer grading. In *International Conference on Artificial Intelligence in Education*, pages 354–365. Springer.
- Longwei Cong, Leon Hammerla, Sonja Hahn, Sebastian Gombert, Hendrik Drachsler, and Ulf Kroehne. 2026. Automatic short answer grading with llms: From memorization to reasoning. In *Proceedings of the LAK26: 16th International Learning Analytics and Knowledge Conference*, pages 75–84.
- Berit K Czinczel, Daniela Fiedler, and Ute Harms. 2025. How do species change over time? designing a hybrid teaching unit on five factors of evolution. *The American Biology Teacher*, 87(2):78–83.
- 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.
- Anna Filighera, Siddharth Parihar, Tim Steuer, Tobias Meuser, and Sebastian Ochs. 2022. Your answer is incorrect... would you like to know why? introducing a bilingual short answer feedback dataset. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8577–8591.
- Scott Frohn, Tyler Burleigh, and Jing Chen. 2025. Automated scoring of short answer questions with large language models: Impacts of model, item, and rubric design. In *International Conference on Artificial Intelligence in Education*, pages 44–51. Springer.
- Sebastian Gombert, Daniele Di Mitri, Onur Karademir, Marcus Kubsch, Hannah Kolbe, Simon Tautz, Adrian Grimm, Isabell Bohm, Knut Neumann, and Hendrik Drachsler. 2023. Coding energy knowledge in constructed responses with explainable NLP models. *Journal of Computer Assisted Learning*, 39(3):767–786.
- Sebastian Gombert, Sonja Hahn, Nico Andersen, Leon Camus, Zhifan Sun, Ngoc Nhu Hao Nguyen, Fabian Zehner, Longwei Cong, Alexander Mehler, and Hendrik Drachsler. 2026a. Rubrics as semantic subspaces: A unified approach to rubric-based constructed response scoring across short answers and essays. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berit Katharina Czinczel, David Bednorz, Marcus Kubsch, Daniele Di Mitri, Knut Neumann, and 1 others. 2026b. Are rubrics all you need? towards rubric-based automatic short answer scoring via guided rubric-answer alignment. In *Proceedings of the LAK26: 16th International Learning Analytics and Knowledge Conference*, pages 272–282.
- Jonas Gwozdz and Andreas Both. 2026. WSE research at BEA 2026 shared task 2: Multi-strategy rubric-based short answer scoring for german: Multi-strategy rubric-based short answer scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Ben Hamner, Jaison Morgan, lynnvandev, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring. <https://kaggle.com/competitions/asap-aes>. Kaggle.
- Marcus Kubsch, Berit Czinczel, Jannik Lossjew, Tobias Wyrwich, David Bednorz, Sascha Bernholt, Daniela Fiedler, Sebastian Strauß, Ulrike Cress, Hendrik Drachsler, and 1 others. 2022. Toward learning progression analytics—developing learning environments for the automated analysis of learning using evidence centered design. In *Frontiers in education*, volume 7, page 981910. Frontiers Media SA.
- Zhaohui Li, Susan Lloyd, Matthew Beckman, and Rebecca J Passonneau. 2023. Answer-state recurrent relational network (AsRRN) for constructed response assessment and feedback grouping. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3879–3891.
- Zhexiong Liu and Jing Zhang. 2026. SDPA at BEA 2026 shared task 2: Efficient LLM fine-tuning for rubric-based short answer scoring. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Jannik Lossjew and Sascha Bernholt. 2025. Digital support custom-made or in moderation—design and evaluation of a digitally supported teaching unit on reaction kinetics and chemical equilibrium. *CHEMKON*, 32(4):122–131.
- OECD. 2023. *PISA 2022 Assessment and Analytical Framework*. PISA. OECD Publishing.
- OECD. 2024. *PISA 2022 Technical Report*. PISA. OECD Publishing.
- Ulrike Padó. 2026. HFT at BEA 2026 shared task 2: Blunt-edge models for hybrid grading. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Ulrike Padó, Yunus Eryilmaz, and Larissa Kirschner. 2023. [Short-answer grading for german: Addressing the challenges](#). *International Journal of Artificial Intelligence in Education*, 34(4):1321–1352.

- Ernesto Panadero and Anders Jonsson. 2013. The use of scoring rubrics for formative assessment purposes revisited: A review. *Educational research review*, 9:129–144.
- Y Malini Reddy and Heidi Andrade. 2010. A review of rubric use in higher education. *Assessment & evaluation in higher education*, 35(4):435–448.
- Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. [Investigating neural architectures for short answer scoring](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.
- Ignacio Sastre, Ignacio Remersaro, Facundo Díaz, Nicolás de Horta, Luis Chiruzzo, Aiala Rosá, and Santiago Góngora. 2026. RETUYT-INCO at BEA 2026 shared task 2: Meta-prompting in rubric-based scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Shashank Sonkar, Kangqi Ni, Lesa Tran Lu, Kristi Kincaid, John S Hutchinson, and Richard G Baraniuk. 2024. Automated long answer grading with ricechem dataset. In *International Conference on Artificial Intelligence in Education*, pages 163–176. Springer.
- Tianqi Wang, Naoya Inoue, Hiroki Ouchi, Tomoya Mizumoto, and Kentaro Inui. 2019. [Inject rubrics into short answer grading system](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 175–182, Hong Kong, China. Association for Computational Linguistics.
- Yuchen Wei, Dennis Pearl, Matthew Beckman, and Rebecca J Passonneau. 2025. Concept-based rubrics improve LLM formative assessment and data synthesis. *arXiv preprint arXiv:2504.03877*.
- Alistair Willis and Aisling Third. 2026. AMATI at BEA 2026 shared task 2: Automatic short answer grading with inductive logic programming and a large language model. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Tobias Wyrwich, Marcus Kubsch, Hendrik Drachsler, and Knut Neumann. 2025. [Tracking students’ progression in developing understanding of energy using ai technologies](#). *Physical Review Physics Education Research*, 21(1).
- Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. 2022. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7).
- 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.

A Appendix

Rank	Team Name	↑ Quadratic Weighted Kappa	Weighted Precision	Weighted Recall	Weighted F1-Score
1	IWM-DKM	.726	.887	.887	.887
2	WSE Research	.717	.882	.883	.883
3	WSE Research	.717	.882	.883	.883
4	IWM-DKM	.710	.880	.881	.881
5	WSE Research	.710	.880	.880	.880
6	WSE Research	.702	.878	.879	.878
7	IWM-DKM	.700	.876	.877	.876
8	IWM-DKM	.698	.875	.877	.876
9	WSE Research	.697	.874	.875	.874
10	IWM-DKM	.684	.869	.870	.869
11	SDPA	.682	.869	.866	.867
12	SDPA	.679	.867	.866	.866
13	RETUYT-INCO	.674	.865	.861	.863
14	RETUYT-INCO	.673	.866	.861	.862
15	RETUYT-INCO	.672	.864	.862	.863
16	RETUYT-INCO	.667	.862	.862	.862
17	ASLAN	.663	.861	.864	.862
18	SDPA	.662	.860	.862	.860
19	SDPA	.656	.857	.856	.857
20	ASLAN	.654	.860	.863	.859
21	ASLAN	.654	.857	.860	.858
22	RETUYT-INCO	.654	.858	.851	.853
23	SDPA	.652	.856	.858	.856
24	ASLAN	.648	.857	.860	.856
25	AMATI	.644	.857	.843	.847
26	AMATI	.644	.857	.843	.847
27	ASLAN	.637	.850	.853	.851
28	Diffuser	.615	.840	.842	.841
29	Afrilan	.546	.814	.803	.807
30	AMATI	.489	.801	.807	.795
31	AMATI	.485	.802	.807	.794
32	AMATI	.485	.802	.807	.794
33	HFT	.477	.786	.793	.787
34	HFT	.465	.781	.789	.783
35	HFT	.435	.766	.770	.768
36	HFT	.424	.761	.766	.763

Table 5: The best five runs per team for the Unseen Answers 2-way track.

Rank	Team Name	↑ Quadratic Weighted Kappa	Weighted Precision	Weighted Recall	Weighted F1-Score
1	IWM-DKM	.550	.813	.818	.815
2	SDPA	.535	.806	.804	.805
3	WSE Research	.533	.806	.811	.808
4	SDPA	.531	.806	.811	.807
5	IWM-DKM	.526	.802	.797	.799
6	WSE Research	.525	.801	.804	.803
7	WSE Research	.510	.799	.806	.800
8	WSE Research	.503	.792	.792	.792
9	WSE Research	.503	.792	.792	.792
10	IWM-DKM	.501	.796	.804	.797
11	RETUYT-INCO	.49	.787	.791	.789
12	HFT	.482	.786	.793	.788
13	HFT	.477	.784	.791	.785
14	HFT	.467	.780	.787	.782
15	SDPA	.461	.776	.781	.778
16	ASLAN	.457	.780	.789	.779
17	ASLAN	.454	.778	.787	.778
18	HFT	.452	.772	.763	.767
19	SDPA	.452	.771	.772	.772
20	ASLAN	.452	.772	.779	.774
21	HFT	.447	.770	.760	.764
22	Afrilan	.437	.766	.757	.761
23	SDPA	.436	.767	.775	.769
24	RETUYT-INCO	.432	.764	.770	.766
25	ASLAN	.417	.770	.780	.765
26	ASLAN	.388	.747	.757	.75
27	RETUYT-INCO	.341	.746	.760	.736
28	AMATI	.289	.704	.714	.708

Table 6: The best five runs per team for the Unseen Questions 2-way track.

Rank	Team Name	↑ Quadratic Weighted Kappa	Weighted Precision	Weighted Recall	Weighted F1-Score
1	IWM-DKM	.796	.781	.782	.780
2	IWM-DKM	.790	.772	.77	.771
3	WSE Research	.790	.773	.774	.773
4	WSE Research	.790	.769	.768	.768
5	WSE Research	.788	.773	.773	.773
6	WSE Research	.784	.765	.765	.765
7	WSE Research	.781	.765	.765	.765
8	IWM-DKM	.781	.779	.767	.77
9	IWM-DKM	.78	.775	.768	.77
10	IWM-DKM	.779	.771	.763	.765
11	SDPA	.776	.775	.763	.766
12	SDPA	.758	.746	.749	.744
13	SDPA	.757	.767	.750	.753
14	ASLAN	.757	.758	.743	.746
15	SDPA	.757	.770	.752	.755
16	SDPA	.756	.760	.755	.756
17	ASLAN	.754	.757	.741	.744
18	AMATI	.749	.760	.736	.739
19	ASLAN	.747	.750	.737	.740
20	ASLAN	.744	.744	.728	.731
21	ASLAN	.739	.740	.725	.728
22	RETUYT-INCO	.729	.728	.728	.728
23	AMATI	.721	.755	.711	.715
24	Diffuser	.698	.712	.703	.705
25	RETUYT-INCO	.695	.701	.702	.702
26	Afrilan	.647	.659	.661	.658
27	AMATI	.614	.662	.655	.650
28	AMATI	.567	.645	.633	.629
29	AMATI	.491	.596	.596	.595

Table 7: The best five runs per team for the Unseen Answers 3-way track.

Rank	Team Name	↑ Quadratic Weighted Kappa	Weighted Precision	Weighted Recall	Weighted F1-Score
1	IWM-DKM	.681	.680	.664	.669
2	WSE Research	.672	.670	.663	.665
3	WSE Research	.653	.664	.662	.663
4	SDPA	.644	.634	.633	.633
5	SDPA	.636	.650	.627	.633
6	IWM-DKM	.635	.657	.640	.644
7	SDPA	.629	.644	.625	.630
8	SDPA	.627	.646	.617	.624
9	WSE Research	.625	.650	.650	.650
10	WSE Research	.625	.655	.647	.648
11	WSE Research	.621	.649	.650	.649
12	SDPA	.621	.630	.628	.628
13	IWM-DKM	.591	.727	.601	.600
14	ASLAN	.579	.653	.587	.593
15	ASLAN	.542	.701	.576	.570
16	IWM-DKM	.541	.738	.576	.563
17	ASLAN	.539	.676	.571	.567
18	ASLAN	.525	.643	.566	.567
19	Afrilan	.523	.607	.608	.607
20	ASLAN	.517	.609	.562	.568
21	AMATI	.394	.525	.525	.520

Table 8: The best five runs per team for the Unseen Questions 3-way track.

HFT at BEA 2026 Shared Task 2: Blunt-Edge Models for Hybrid Grading

Ulrike Padó

Hochschule für Technik (HFT) Stuttgart

Schellingstr. 24

70174 Stuttgart, Germany

ulrike.pado@hft-stuttgart.de

Abstract

Open-source LLMs with simple, zero-shot prompts are at best middling graders on the BEA 2026 Automated Grading Shared Task – blunt-edge models, in fact. However, they are good enough to support human graders and save them time. We demonstrate the application of a hybrid grading approach that first transparently defines the success criteria and then pairs a zero-shot LLM grader with human review. The hybrid approach outperforms the LLM grader on its own and has the added advantage of keeping the human in the loop.

1 Introduction

Short answer questions are a welcome opportunity for educators to probe their students' ability to explain or analyze a problem. By phrasing an answer in their own words, the students prove their factual knowledge, show their thought processes and also demonstrate their language ability in language instruction settings. However, manual grading of short answer questions, while yielding useful insights into common misconceptions or errors, is time-consuming. Therefore, many attempts at automating short answer grading have been made over the years, and have profited strongly from the availability of common benchmark data.

The BEA 2026 Short-Answer Grading Shared Task (Gombert et al., 2026) offers a new data set for German rubric-based scoring. We approach this data with LLM grading models that deal gracefully with rubric instructions. We specifically choose open-source models to ensure independence of commercial offerings: These are prone to hidden changes or complete retraction and therefore endanger mid- to long term replicability and pose risks to data privacy in practical use.

Also, we intentionally limit ourselves to the "blunt edge" of research: Instead of creating an optimized "cutting-edge" model, we stress low cost

and broad re-usability for other tasks by evaluating a simple zero-shot approach using open-source LLMs and standardized prompts from the literature. Our blunt-edge model performs close to more sophisticated offerings in the information-poor Unseen Question setup, but clearly lags behind approaches that integrate the available information in the Unseen Answer setting (see Section 3.3).

To address imperfect model performance, we propose using a hybrid human-machine grading process that integrates human decision making with automated grading in such a way that the most reliable machine grade predictions are retained and the less reliable ones revised manually. The performance requirements of the process are defined by the educator beforehand (Padó et al., 2024). The simulated output of the hybrid process is competitive with cutting-edge models, especially when no sample answers are available (Section 4).

Finally, we argue that partial (or full) human review of the proposed model grades not only allows the hybrid process to reach the pre-defined performance requirements; it also gives the educator insights into the student data. In fact, human oversight over grading decisions is crucial for high-stakes systems in practical use in the European Union: The recent EU AI Act requires extensive risk management and documentation of any AI system that grades student submissions without human intervention.¹ In contrast, this explicitly does not apply to systems that just prepare data for a human evaluation process.²

We now describe our approach in more detail, starting with the model and prompt selection for our blunt-edge model on the Shared Task trial data, its performance in the competition proper and the expected outcome of the hybrid process on the (unseen) Shared Task training data.

¹Such systems are classed as high-risk systems in the AI Act, Chapter 3, Section 1, Article 6(3) and Annex III, (3)

²Cf. AI Act, Chapter 3, Section 1, Article 6(3d)

2 Related Work

Short-Answer Grading is a well-established task (see the overviews of [Burrows et al., 2015](#); [Bai and Stede, 2023](#)). While most of the research is on English data, several German data sets have been collected over the last 15 years, e.g., CREG ([Meurers et al., 2011](#)), CSSAG ([Padó and Kiefer, 2015](#)), ASAP-DE ([Horbach et al., 2018](#)) and SAF-DE ([Filighera et al., 2022](#)). However, especially the older data sets are limited in size, making the BEA 2026 Shared Task 2 data a welcome addition.

Along with every subfield of Natural Language Processing, Short-Answer Grading for German has seen a surge of experimentation with LLMs. [Filighera et al. \(2022\)](#) used a Transformer model with a trained classification head on the SAF-DE data and saw up to 85% prediction Accuracy in cases where the question had been seen during training (but the test answers had not) and 55% for unseen questions. With a similar approach, [Padó \(2016\)](#) achieved 84.4% prediction Accuracy on the CSSAG data on seen questions and 69% on unseen questions. These numbers are similar to non-neural benchmarking results for CREG (85% seen-question Accuracy, [Meurers et al. 2011](#)).

Using one-shot prompting that provided the LLM with the correct reference answer and the expected range of points, [Speiser and Weng \(2024\)](#) were able to vastly improve Accuracy on unseen questions from CSSAG to 87%. Note that [Metzler et al. \(2024\)](#), working on English data and a small German student answer data set in the statistics domain, recommend prompting using rubrics over n-shot answer examples, because they find that while n-shot prompting improve results, it also leads to inconsistent predictions.

In sum, prompting LLMs is a promising approach to solving the automatic grading task, especially since they can make use of the information in rubrics without further pre-processing, and appear to show more consistent performance with rubrics. However, the literature also shows that providing as much information as possible in the form of reference answers, sample answers or the expected point distribution improves prediction performance.

Regarding the human-machine hybrid grading strategy, a number of proposals have been made over time to integrate machine predictions with human grading or revision, in order to lighten the human workload. Some approaches cluster student answers to reduce the need for labeling to one in-

stance per cluster ([Basu et al., 2013](#); [Zehner et al., 2016](#)), with the advantage of allowing teachers insight into their students' frequent error patterns and potentially giving feedback per cluster.

Human review of some part of the machine predictions without clustering, based on the confidence of the grading models and their strengths and weaknesses, is also proposed by [Schneider et al. \(2023\)](#) or [Kwako et al. \(2026\)](#). Alternatively, [Speiser and Weng \(2024\)](#), like [Vittorini et al. \(2021\)](#), advocate for human-machine dual grading with revisions only in case of disagreement, which sacrifices workload reduction for full human review and improved correctness. However, [Vittorini et al. \(2021\)](#) still find a 40% speedup for reviewing over grading from scratch.

An investigation of very restricted, open-source LLMs to solve a Shared Task was undertaken by [Góngora et al. \(2025\)](#) at the BEA Shared Task 2025 (Pedagogical Ability Assessment of AI-powered Tutors): They only looked at models with 1b parameters or less to demonstrate the limits of what is possible with restricted access to computation resources. [Padó \(2026\)](#) approaches the 2025 Shared Task with the same blunt-edge, hybrid grading process methodology applied here, and finds that the hybrid process can be successfully applied to reach a pre-defined Accuracy threshold.

3 Shared Task: Grading Models

3.1 Method

We experiment with the open-source LLMs available to us for research purposes through Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen. We work with the seven text-based models that responded in less than 30s per API request and consistently produced the standardized JSON output format needed for post-processing. These were text-based models gpt-oss³, qwen3-32b⁴, gemma-3-27-it⁵, mistral-large-3-675b-instruct-2512⁶ and llama-3.3-70b-instruct⁷, specialized reasoning model qwen3-30b-a3b-thinking-2507⁸

³<https://huggingface.co/openai/gpt-oss-120b>

⁴<https://huggingface.co/Qwen/Qwen3-32B>

⁵<https://huggingface.co/google/gemma-3-27b-it>

⁶<https://huggingface.co/mistralai/Mistral-Large-3-675B-Instruct-2512>

⁷<https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>

⁸<https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507>

and llama-3.1-sauerkrautlm-70b-instruct⁹, which was fine-tuned on German-English data. All models were run at a temperature of 0.01, so in conservative mode. We used the RAGAS¹⁰ libraries for implementation.

The second parameter in our setup is the nature of the prompt. We use four prompts in a zero-shot setting (i.e., the model receives instructions, but no examples), cf. Padó (2026). Our *Baseline* prompt just instructs the LLM to return "Correct" if the student answer is correct given the rubric, and "Incorrect" otherwise. Next, a prompt proposed by Fan et al. (2025) in the BEA 2025 tutor evaluation task goes one step further and explains the task context (a student answering a question and receiving a grade). An alternative prompt specifies a *Role* for the LLM (it simulates an experienced STEM teacher). Finally, we also use a prompt template by Wang et al. (2024), which instructs the model to simulate a committee of experts solving the task (the prompt was shortened to one sample conversation). All the prompts were in German (translated from the originals as needed) and can be found in the Appendix. With this set of progressively more complex prompts, we want to probe how much information the LLMs need to grade well.

3.2 Model and Prompt Selection

We evaluate our seven models and four prompts on the Shared Task trial data in the two-way setting (labels "Correct" and "Incorrect"). Our zero-shot approach mimics the Unseen Question task. Table 1 shows the results. As expected, the sparse Baseline **prompt** rarely yields best results, but neither does the very complex Wang et al. prompt (Mistral, SauerkrautLM and Gemma even do much worse than for the other prompts). Instead, models do best with the shorter Fan et al. and Role prompts that provide some context to the task but are essentially simple. Variation in our LLMs' predictions, accounts for as much as 0.025 points QWK difference across five runs for Llama+Fan et al. (max. 0.49, min. 0.46, average 0.48) and 0.016 points QWK across five runs for GPT-OSS+Fan et al. (max. 0.48, min. 0.46, average 0.48). Given this observation, performance is surprisingly consistent across the Baseline, Fan et al. and Role prompts and across different models. This is positive because it shows a certain robustness to model

identity and details of prompting that promises that performance will carry over to other configurations.

In an unplanned experiment, a correction of the training and trial data for the 2-way task gave insight into the importance of covering the whole answer space with rubrics. When the initial data set that had only the rubrics for "Incorrect" and "Correct" was replaced with a revised data set that also contained the "Partially Correct" rubrics (to count as "Incorrect"), grading performance went up by around 0.03 points QWK across all models and prompts. This was carried by improved Precision for the "Correct" predictions at almost unchanged Precision for label "Incorrect"; clearly, some of the "Partially Correct" answers had initially been erroneously identified as "Correct" due to the partial overlap with the "Correct" rubrics.

3.3 Results on Shared Task Test Set

For evaluation on the test data, we chose the best-performing models and submitted several runs each in order to account for prediction variation. The best-performing and most consistent models are GPT-OSS and Llama. We paired both models with the Fan et al. prompt, the simplest successful prompt.

We submitted model predictions for the 2-way Unseen Question and Unseen Answer tasks. Since we use a zero-shot setting, we expect to see the same performance on both test tasks as on the trial data reported above - around 0.48 QWK.

Table 2 shows that this expectation was borne out by the GPT-OSS+Fan et al. model. Llama+Fan et al. did somewhat worse, although performance on the trial data had generally been similar. Still, the blunt-edge models again show remarkable robustness, this time across data sets, and achieve 5th place (out of 9) in the 2-way UQ task.

The competition results on the test data show that the largest performance gains of the cutting-edge models are in the Unseen Answer setting, where task-specific information is available (Gombert et al., 2026). Here, the blunt-edge model places last. Even though the rubrics provide it with some task-specific information, more knowledge about the task in the shape of previously graded answers remains hugely beneficial – a familiar pattern in ASAG independent of grading algorithm. We see this situation as a trade-off: If data, development time and computing resources are available and models can be re-used often, the development of a cutting-edge approach is clearly worth it. However,

⁹<https://huggingface.co/VAG0solutions/Llama-3.1-SauerkrautLM-70b-Instruct>

¹⁰<https://docs.ragas.io/en/latest/>

Model	Baseline	Fan et al.	Role	Wang et al.
GPT-OSS	0.4772	0.4833	0.4759	0.4846
Llama	0.4772	0.4927	0.4846	0.4538
Mistral	0.4632	0.4366	0.4632	0.3428
Qwen3-32b	0.4621	0.4840	0.4832	0.4207
Qwen3-thinking	0.4535	0.4335	0.4602	0.4595
SauerkrautLM	0.4464	0.4453	0.4171	0.3290
Gemma	0.4056	0.4181	0.4351	0.3784

Table 1: QWK for LLMs and four prompts on the trial data, two-way Unseen Question task (best result in bold).

Model	2-way UQ			2-way UA		
	QWK	F_1	Pos.	QWK	F_1	Pos.
IWM-DKM	0.55	0.815	1	0.726	0.887	1
GPT-OSS	0.482	0.788	5	0.477	0.783	9
Llama	0.452	0.767	–	0.435	0.766	–

Table 2: QWK and weighted F_1 for the two-way Unseen Question (UQ) and Unseen Answer (UA) test data. Task winner and our models (Fan et al. prompt), showing the best submitted result and the team rank (out of nine).

in many teaching settings questions change from exam iteration to exam iteration with no sample answers available; in these cases, the blunt-edge models are a viable, low-cost alternative.

4 Hybrid Grading

As expected, the zero-shot open-source models on their own do not perform competitively on the Shared Task test data, especially in the information-rich Unseen Answer setting. However, we argue that they are fully sufficient to support hybrid human-machine grading when following our four-step grading process (cf. Padó et al. 2024). The process guides an educator through model selection and hybrid grading, by identifying the concrete grading model’s “preferred” class, which it predicts best. Accepting predictions for this class and reviewing all predictions of the other class(es) reduces human workload and significantly improves grading results over machine grading alone. This model-based selection of presumably reliable predictions precludes the need for confidence scores for each grade, which may not be available.

We now step through the process and apply it to the (unseen) Shared Task training data.

Step 1: Define Quality Threshold Since the maximum acceptable error for the output of the grading process can vary for each usage context, the hybrid grading process calls for defining it explicitly first. Published short-answer corpora show 10-15% disagreement (lower for higher stakes)

among human graders (Mieskes and Padó, 2018; Nazaretsky et al., 2022). Therefore, we accept 15% error as the expected human performance.

Step 2: Choose Data Set The next task is to identify data from one’s own setting with existing human grades to test the grading models on. On the basis of this evaluation, the final grading setting for the hybrid grading process is chosen. This step recognizes that model performance can vary widely between data sets and that benchmark performance cannot be expected to carry over to different settings. We use the Shared Task trial data.

Step 3: Choose Grading Model We use our results for the trial data from 3.2. At this step in the process, however, we drill further down into the two best models’ performance to identify their “preferred” classes and class-specific performance. We look for high Precision on at least one class – optimally, at least the accepted error threshold for the task (cf. Step 1) to ensure acceptable error for each class. Our two best models are close to this threshold: GPT-OSS+Fan et al. shows $Prec_{Incorrect} = 83.00$ and Llama 3.3+Fan et al. has $Prec_{Incorrect} = 84.3$.

Step 4: Decide on Use Given the results in Step 3, we decide to use Llama 3.3+Fan et al. to grade the Shared Task training data, which are unseen to our zero-shot model. We will accept any “Incorrect” grades and assume that human review fixes all errors in the “Correct” grade predictions.

On the 7072 answers in the training set, the model made 5103 predictions of class "Incorrect" (to be accepted as-is) and 1969 predictions of class "Correct" (to be revised). This means a reduction in human workload by 71%.

The gold labels show that 840 predictions of "Incorrect" were in fact wrong. If the revised labels are assumed to be perfect, this makes for a remaining error of 12%, well within our margins. Weighted F_1 for the hybrid process would be 0.896 in this best-case scenario, which is higher than the best reported F_1 score in the competition.

If we (more realistically) assume that the human expert makes evenly-distributed errors for 15% of the 1969 labels during revision, missing some true errors and marking some truly correct answers as incorrect, overall error for the hybrid process is 16% (still close to our threshold) and weighted F_1 drops to 0.828 – eighth place in the Unseen Answers track, and first in the information-poor Unseen Questions track. This makes the hybrid process a clear recommendation for cases where little is known in advance about the questions and answers: In such cases, the simple blunt-edge model saves development time and resources, while the hybrid setup ensures competitive performance and allows the human grader insights into students' answer patterns.

5 Discussion and Conclusions

We have presented our contribution to the BEA 2026 Automated Grading Shared Task: A blunt-edge, but replicable and relatively low-cost LLM grader that relies on off-the-shelf open source LLMs and simple zero-shot prompts.

While the grader holds its own in the information-poor Unseen Questions task, it clearly lags behind the cutting-edge models in the Unseen Answer task. However, we argue that it is good enough to be combined with human judgments in a hybrid grading process which accepts the model grades for the class predicted with higher Precision and presents the other grades to human review. This hybrid process even outperforms the cutting-edge models in simulations on unseen data, especially in information-poor settings.

Note that the hybrid strategy is applicable to any grading model, and further performance gains are possible with better machine graders. For any grading task that has little previously-annotated training data, we still explicitly recommend the use of blunt-

edge models, however. The use of off-the-shelf open source LLMs ensures data privacy, replicability and reduction of the tasks' energy footprint, and blunt-edge performance is robust in information-poor settings.

In any case, the manual revision step allows the human graders insight into answer patterns that are not available in fully automated grading.

Importantly, the hybrid grading process has a bias towards whichever label is predicted with less error by the grading model and therefore accepted without review. This is often the majority class in the training data, especially for trained or fine-tuned models. For the Shared Task data, this class is "Incorrect". This means that all the remaining error after application of the hybrid process is concentrated in cases where the machine erroneously predicts a student to be wrong. In a summative setting, this is a worst-case scenario. In a formative setting, it can be acceptable depending on the amount of feedback a student receives and the opportunities they have to clear up confusion caused by inaccurate machine grades.

Note, however, that depending on the data set and grading model, the process bias can well be inverted (towards "Correct"). This is why we strongly advocate for experimenting with the grading model on realistic data before choosing the manual-machine balance for the task in question. For example, educators could choose complete manual review for high-stakes tests or in case of poor model performance on their data, or on the opposite end of the spectrum skip manual review completely if grading a low-stakes feedback activity and using a reliable model with an acceptance bias for "Correct". In fact, the hybrid process has the advantage of allowing this human-machine balancing on an individual basis while taking into account the testing situation, the expected reliability of the machine grades, the available human work time and individual preferences for manual oversight.

Naturally, to fully avoid unfair bias in all situations, all machine grades have to be revised every time. This can still be worthwhile, since [Vittorini et al. \(2021\)](#) find a time saving of 40% for revision over grading from scratch. Complete revision also ensures that the requirements of the EU AI Act for the use of AI in grading are met. For both these reasons, full review is recommended in high-stakes situations.

Limitations

Our approach has been simulated for different data sets, but a full trial with real human annotators is still outstanding. To our understanding, the partial review process does not satisfy the requirements for low-risk AI use according to the EU AI Act. For this reason, additional experiments on the time savings of full vs. partial review should also be done. No additional inquiries into the effect of adding more task information (for example through few-shot prompting) in the Seen Question task were made.

Ethics

Our approach minimizes, but does not eliminate, the energy footprint of LLM usage. There is a possible bias in the resulting grades since errors concentrate in one predicted class. Also, there are currently no checks for other machine grading biases, e.g., against students who are not native speakers, cf. Padó et al. (2024)).

Acknowledgments

The author gratefully acknowledges the LLM hosting services granted by the KISSKI project of Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG). Many thanks also to two anonymous reviewers for their constructive comments.

References

- Xiaoyu Bai and Manfred Stede. 2023. [A survey of current machine learning approaches to student free-text evaluation for intelligent tutoring](#). *Int. J. Artif. Intell. Educ.*, 33(4):992–1030.
- 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.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25:60–117.
- Yuming Fan, Chuangchuang Tan, and Wenyu Song. 2025. [BJTU at BEA 2025 shared task: Task-aware prompt tuning and data augmentation for evaluating AI math tutors](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1073–1077, Vienna, Austria. Association for Computational Linguistics.
- Anna Filighera, Siddharth Parihar, Tim Steuer, Tobias Meuser, and Sebastian Ochs. 2022. Your answer is incorrect... would you like to know why? Introducing a bilingual short answer feedback dataset. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8577–8591.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, and Hendrik Drachsler. 2026. Report on the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Santiago Góngora, Ignacio Sastre, Santiago Robaina, Ignacio Remersaro, Luis Chiruzzo, and Aiala Rosá. 2025. [RETUYT-INCO at BEA 2025 shared task: How far can lightweight models go in AI-powered tutor evaluation?](#) In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1135–1144, Vienna, Austria. Association for Computational Linguistics.
- 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.
- Alexander Kwako, Susan Lottridge, and Christopher Ormerod. 2026. [Using confidence modeling to optimize overall score quality in hybrid scoring systems](#). *Educational Measurement: Issues and Practice*, 45(2):e70019.
- Tim Metzler, Paul G. Plöger, and Jörn Hees. 2024. Computer-assisted short answer grading using large language models and rubrics. In *INFORMATIK, Lecture Notes in Informatics (LNI)*. Gesellschaft für Informatik.
- Detmar Meurers, Ramon Ziai, Niels Ott, and Stacey Bailey. 2011. Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *International Journal of Continuing Engineering Education and Life-Long Learning*, 21(4):355–369.
- Margot Mieskes and Ulrike Padó. 2018. Work smart - reducing effort in short-answer grading. In *Proceedings of the 7th Workshop on NLP for Computer Assisted Language Learning*, pages 57–68.
- Tanya Nazaretsky, Moriah Ariely, Mutlu Cukurova, and Giora Alexandron. 2022. Teachers’ trust in AI-powered educational technology and a professional development program to improve it. *British Journal of Educational Technology*, 53(4):914–931.
- Ulrike Padó. 2016. Get semantic with me! The usefulness of different feature types for short-answer grading. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2186–2195.

Ulrike Padó. 2026. Hybrid evaluation of tutor dialogues. In *Agentic learning Ecosystems and ITS, 22nd International Conference (ITS 2026)*. Springer Nature Switzerland. To appear.

Ulrike Padó, Yunus Eryilmaz, and Larissa Kirschner. 2024. [Short-answer grading for German: Addressing the challenges](#). *International Journal of Artificial Intelligence in Education*, 34(4).

Ulrike Padó and Cornelia Kiefer. 2015. Short answer grading: When sorting helps and when it doesn't. In *Proceedings of the Workshop on NLP for Computer-Aided Language Learning*, pages 42–50, Vilnius, Lithuania.

J. Schneider, R. Richner, and M. Riser. 2023. [Towards trustworthy autograding of short, multi-lingual, multi-type answers](#). *Int J Artif Intell Educ* 33, 33:88–118.

Sebastian Speiser and Annegret Weng. 2024. Enhancing short answer grading with OpenAI APIs. In *IEEE International Conference on IT in Higher Education and Training (ITHET)*.

Pierpaolo Vittorini, Stefano Menini, and Sara Tonelli. 2021. An AI-based system for formative and summative assessment in Data Science courses. *International Journal of Artificial Intelligence in Education*, 31:159–185.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024. [Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 257–279, Mexico City, Mexico. Association for Computational Linguistics.

Fabian Zehner, Christine Sälzer, and Frank Goldhammer. 2016. Automatic coding of short text responses via clustering in educational assessment. *Educational and psychological measurement*, 2(76):280–303.

A Appendix: Prompts

Baseline "Bewerte mit 1, wenn die Antwort gegeben die Bewertungsvorschrift richtig ist. Sonst bewerte mit 0."

Fan et al. "Ein Schüler beantwortet eine Frage. Deine Aufgabe ist es, zu bewerten, ob die Antwort auf die Frage richtig ist. Die Bewertungsvorgaben für Richtig und Falsch liegen dir vor. Deine Aufgabe ist, zu bewerten, ob die Antwort gegeben die Bewertungsvorschriften richtig ist. Bewerte mit 1, wenn die Antwort richtig ist. Sonst bewerte sie mit 0."

Role "Du bist ein erfahrener Lehrer in den Naturwissenschaften. Du bewertest eine Schülerantwort auf eine Frage. Du hast Bewertungsrichtlinien gegeben. Bewerte die Antwort mit 1, wenn sie den Richtlinien entsprechen. Sonst bewerte sie mit 0."

Wang et al. "Wenn du eine Aufgabe bekommst, beginne damit, die Beteiligten zu identifizieren, die zur Lösung beitragen werden. Dann starte einen mehrstufigen Zusammenarbeitsprozess, bis eine endgültige Lösung gefunden ist. Die Beteiligten geben dir kritische Rückmeldungen und detaillierte Vorschläge, wenn nötig. Hier sind einige Beispiele:

Beispiel-Aufgabe 1: Verwende Zahlen und grundlegende Rechenoperationen (+ − * /), um 24 zu erhalten. Du musst alle Zahlen verwenden, und jede Zahl darf nur einmal genutzt werden. Eingabe: 6 12 1 1 Beteiligte: Du (KI-Assistent); Mathematik-Experte Start der Zusammenarbeit! Mathematik-Experte: Lass uns die Aufgabe genau analysieren. Du musst sicherstellen, dass du die Anforderung erfüllst: Du musst genau die vier Zahlen (6 12 1 1) verwenden, um 24 zu bilden. Um 24 zu erreichen, kannst du an die häufigen Teiler von 24 denken, wie 4, 6, 8 oder 3, und versuchen, diese zunächst zu konstruieren. Außerdem solltest du potenzielle Summen in Betracht ziehen, die 24 ergeben, wie zum Beispiel 12 + 12. Du (KI-Assistent): Danke für die Hinweise! Hier ist eine erste Lösung: $(12 / (1 + 1)) * 6 = 24$ Mathematik-Experte: Lass uns die Antwort Schritt für Schritt überprüfen. $(1 + 1) = 2$, $(12 / 2) = 6$, $6 * 6 = 36$ – das ist nicht 24! Die Antwort ist falsch. Kannst du das durch andere Kombinationen korrigieren? Bitte vermeide ähnliche Fehler. Du (KI-Assistent): Danke, dass du den Fehler aufgezeigt hast. Hier ist eine überarbeitete Lösung, bei der berücksichtigt wird, dass 24 auch durch 3×8 erreicht werden kann: $(6 + 1) \times (12 / 4) = 24$. Mathematik-Experte: Lass uns prüfen, ob die Berechnung stimmt. $(6 + 1 + 1) = 8$, $12 / 4 = 3$, $8 \times 3 = 24$. Die Rechnung ist korrekt, aber du hast die Zahlen 6, 1, 1, 12 und 4 verwendet – das entspricht nicht der Eingabe 6, 12, 1, 1. Kannst du vermeiden, eine Zahl zu verwenden, die nicht in der Eingabe enthalten ist? Du (KI-Assistent): Du hast recht. Hier ist eine überarbeitete Lösung, bei der berücksichtigt wird, dass 24 auch durch 12 + 12 erreicht werden kann, und ohne zusätzliche Zahlen: $6 \times (1 - 1) + 12 = 24$. Mathematik-Experte: Lass uns die Antwort nochmal überprüfen. $1 - 1 = 0$, $6 \times 0 = 0$, $0 + 12$

= 12. Du bist ganz nah dran. Hier ein Tipp: Versuche, „1 – 1“ durch „1 + 1“ zu ersetzen. Du (KI-Assistent): Klar, hier ist die korrigierte Antwort: $6 \times (1 + 1) + 12 = 24$ Mathematik-Experte: Lass uns die Lösung überprüfen. $1 + 1 = 2$, $6 \times 2 = 12$, $12 + 12 = 24$. Du hast die Zahlen 1, 1, 6, 12 verwendet – das entspricht genau der Eingabe 6, 12, 1, 1. Alles passt! Zusammenarbeit beendet! Endgültige Antwort: $6 \times (1 + 1) + 12 = 24$ Identifiziere nun die Beteiligten und lass sie die folgende Aufgabe schrittweise gemeinsam lösen.

Ein Schüler beantwortet eine Frage. Die Bewertungsrichtlinien für Richtig und Falsch liegen dir vor. Bewerte die Antwort mit 1, wenn sie den Richtlinien entsprechen. Sonst bewerte sie mit 0."

ASLAN at BEA 2026 Shared Task 2: Voting Across Scoring Paradigms

Marie Bexte¹, Yuning Ding^{2,3}, Josef Ruppenhofer¹, Nils-Jonathan Schaller^{2,3},
Daniel Mora Melanchthon^{2,3}, Torsten Zesch¹, Andrea Horbach^{2,3},

¹CATALPA, FernUniversität in Hagen, Germany,

²Leibniz Institute for Science and Mathematics Education, Kiel, Germany

³Kiel University, Germany

Abstract

This paper describes the ASLAN system contribution to the BEA 2026 Shared Task on rubric-based short answer scoring for German (Gombert et al., 2026). We investigate three complementary modeling paradigms: similarity-based scoring, instance-based classification, and rubric-prompted large language models (LLMs). For the *unseen answers* track, where test answers belong to prompts observed during training, we compare question-specific and generic scoring models as well as ensemble variants. For the *unseen questions* track, where models must generalize to previously unseen prompts, we primarily rely on zero-shot LLM-based scoring using the scoring rubrics. Our experiments show that similarity-based models outperform instance-based models and LLM-based models in the *unseen answers* setting. In addition, we find that ensemble methods improve robustness over individual models.

1 Introduction

Automated Short Answer Scoring (SAS) is the task of automatically judging the conceptual correctness of learner answers in an educational context (Bai and Stede, 2023; Bexte et al., 2024).

While traditional SAS systems often rely on the comparison with a reference answer or learning properties of a correct or incorrect answer from training instances for a specific question (Bexte et al., 2023, 2022), this Shared Task aims at leveraging the scoring rubrics provided for individual answers as the basis for determining the correctness of an answer - a so far underexplored variant of SAS, that is especially relevant in those cases where no training data for a specific question is provided. The shared task consists of two tracks: *unseen answers* where the test data contains new answers to questions already seen during training and *unseen questions* with completely new questions not covered in the training material. For

both tracks, there is a 3-way condition and a 2-way condition. In the 3-way setup, answers are labeled *correct*, *partially correct* or *incorrect*. The 2-way condition subsumes the *partially correct* and *incorrect* answers in a joint group of *incorrect* answers.

Our system compares several approaches: In the *unseen answers* track, we use similarity-based approaches, relying on the semantic similarity to a reference answer, instance-based approaches that learn properties of correct and incorrect answers to the same question, and LLM-based approaches leveraging generative AI and using the scoring rubric in the prompt. Only the third approach is directly applicable to the *unseen questions* track data. We further experiment with ensembles of classifiers, performing majority voting to determine the final label.

2 Datasets

The provided training dataset consists of a total of 7899 answers, split into train and trial data and covering a total of 78 individual questions. We deemed that appropriate to train question-specific models. All answers came from a real-world ITS system, thus label distributions were unsurprisingly uneven, even including a small number of answers with only correct or only incorrect answers in the dataset.

3 Method

Our system explores a number of complementary approaches: For the *unseen answers* track, we compare both instance-based and similarity-based approaches (Horbach and Zesch, 2019). For the *unseen questions* track, we experiment with approaches using generative LLMs.

In all experiments presented here, we use the train-trial data split provided by the Shared Task organizers. We evaluate using quadratically weighted kappa (QWK) where - although we in some in-

stances train one model per question - kappa values are computed across the whole trial data (instead of computing QWK prompt-wise and then averaging these individual values).

3.1 Similarity-Based Scoring

We tested two similarity-based scoring setups, Label-Aligned Similarity Learning (**LaSiLearn**) (Bexte et al., 2022) and **SetFit** (Tunstall et al., 2022). Both approaches are designed to work with SBERT (Reimers and Gurevych, 2019) base models. We ran both LaSiLearn and SetFit with three different multilingual base models, namely *paraphrase-multilingual-mpnet-base-v2* (**mpnet**), *paraphrase-multilingual-MiniLM-L12-v2* (**mini-lm**), and *distiluse-base-multilingual-cased-v1* (**distil**). All models were used via Hugging Face.

LaSiLearn LaSilearn adapts SBERT models so that the cosine similarity of answer embeddings reflects the similarity of the scores of two answers. For training supervision, pairs of two answers are built from the training data and assigned a similarity label that reflects whether the two answers in a pair share the same (similarity label 1) or a different (similarity label 0) score. With a fine-tuned model, test answers can then be compared to training answers. The score for which training answers have the highest average similarity to a test answer is then used as the prediction. For example, if the average similarity of a test answer is .4 for comparing it to *incorrect* training answers, .6 for *partially correct* training answers and .8 for *correct* training answers, the model would predict the answer to be *correct*. LaSiLearn inference is also possible directly with the pretrained model, without any fine-tuning. Thus, we also test performance of the pretrained base models. As previous work had shown that for pretrained models it is best to predict the score of the most similar answer (Bexte et al., 2022, 2023), we adopt this strategy as well.

SetFit SetFit is a partially similarity-based scoring approach designed for smaller volumes of training data. First, the SBERT model is tuned as described for LaSiLearn in the previous section. Thereby, the embedding space becomes more representative of the answer space. In a second training loop, a classification head is then added to the model and fine-tuned using the same training data that was used for the similarity-based training. At

Task	mpnet	mini-lm	distil	vote	vote-sim
2-way					
LaSiLearn _{pre}	.50	.45	.50	.51	.49
LaSiLearn	.59	.61	.59	.63	.60
SetFit	.65	.62	.62	.65	–
3-way					
LaSiLearn _{pre}	.59	.54	.61	.61	.58
LaSiLearn	.71	.71	.71	.74	.75
SetFit	.70	.68	.71	.70	–

Table 1: QWK results for similarity-based scoring on trial data. LaSiLearn_{pre} indicates direct use of the pre-trained base model without any fine-tuning.

inference, predictions are made using this classification head.

Hyperparameters For both LaSiLearn and SetFit, we set the same hyperparameters. We train with a batch size of 16 for at most 1500 steps, evaluating performance on validation data every 100 steps. The model that performs best on the validation data is kept for final evaluation on the test data. 10% of the training data for each prompt were set aside as validation data.

Results Table 1 shows performance of the similarity-based models on the trial data. Since we ran both scoring methods with three separate multilingual base models, we can aggregate their predictions via voting. For both LaSiLearn and SetFit, we performed majority voting (**vote** in Table 1). In the case of LaSiLearn, we also ran voting based on the similarities the model found. We took the prediction of the model that based its prediction on the highest similarity value (**vote-sim** in Table 1). For example, if one model predicted an answer to be *partially correct* based on a similarity of .78 to the reference answers and another said it would be *incorrect*, but based this on a similarity of .80, we adopt the prediction of the model that found the higher similarity and thus predict the answer to be *incorrect*.

In 2-way scoring, SetFit slightly outperforms LaSiLearn. The mpnet base model performs best, on par with a voted average of the three base models. For LaSiLearn, voting increases performance from .61 QWK to .63. On the 3-way data, LaSiLearn performs better than SetFit. Again, performance is pushed by voting, from .71 QWK to .75 with similarity-based voting. For SetFit, the distil base model performs best, outperforming the voted average.

Based on the results, we chose the following configurations for our submission on the final test data:

- 2-way LaSiLearn: vote
- 3-way LaSiLearn: vote-sim
- 2-way SetFit: mpnet
- 3-way SetFit: distiluse

3.2 Instance-Based Scoring

In the instance-based scoring approach, we treat the task as a supervised classification problem, where models learn to score the correctness of answers from lexical material in the training data. The results are shown in Table 2.

3.2.1 Shallow Classifiers

As a baseline, we trained different shallow classifiers (Random Forest (RF), Support Vector Classifier (SVC), Logistic Regression (LR)) with unigram and bigram features. In Table 2, we only report LR, as this model performed best. We used the scikit-learn (Pedregosa et al., 2011) implementation and apart from setting *max_iter* to 1000 left all parameters at their respective default values.

3.2.2 Transformer Models

We fine-tuned three distinct pre-trained transformer-based models. All models were used via Hugging Face.

- BERT: The standard multilingual BERT base model. (Devlin et al., 2019)
- German-BERT: A BERT model specifically pre-trained on large-scale German corpora.¹
- G-EdSciBERT: A specialized German model focused on educational and scientific domains. (Latif et al., 2024)

Experimental Setup All three models were fine-tuned using the same hyperparameter configuration to ensure a fair comparison. We utilized a learning rate of $2e - 5$, a batch size of 7, and trained for 6 epochs. The models were trained on the provided training set without validation and evaluated on the trial data to determine their performance for the 2-way and 3-way classification tasks.

Results Table 2 summarizes the performance of these models on the trial data, reported using the Quadratically Weighted Kappa (QWK) metric. The

¹<https://huggingface.co/google-bert/bert-base-german-cased>

Task	LR	BERT	German BERT	G-SciEdBERT
2-way	.52	.47	.58	.62
3-way	.63	.60	.65	.67

Table 2: QWK results for instance-based models on trial data.

trial results demonstrate that G-SciEdBERT consistently outperforms the other models on both tasks, achieving QWK values of 0.62 for 2-way classification and 0.67 for 3-way classification. This suggests that the domain-specific pre-training on German scientific and educational texts provides a significant advantage for the SAS task.

Prompt-Specific vs. Generic Training We further investigated the impact of prompt-specific training compared to a generic model approach. In the Generic Model (GM) setup, a single model is trained on the entire pooled dataset. In contrast, we trained individual Prompt-Specific Models (PSM) for each unique question to capture prompt-specific nuances. Our comparison, unsurprisingly, revealed a performance trade-off with the volume of available training data: PSMs have a comparative performance to GMs when the training set for a given prompt exceeds 100 instances. In this range, the PSM has a 52.2% probability of outperforming the GM. When the sample size is small, the GM tends to be more robust. The shared knowledge across prompts in the GM yields a higher average Kappa value compared to PSMs, which likely suffer from overfitting on limited instances. Based on the results, we chose generic training of G-SciEdBERT for our submission on the final test data.

Augmented Input We also experimented with augmenting the training input by concatenating the question text and the scoring rubric to the student answers. We hypothesized that this would provide the model with a better basis for determining correctness. However, this approach did not yield a consistent improvement, potentially due to increased noise or input length limitations.

3.3 LLM-based Scoring

Both closed- and open-source models were evaluated in a zero-shot setting, following the same general prompting strategy: each model received the question, the scoring rubric, and the student answer, and was instructed to predict a score label.

Task	GPT5-mini	Phi-4	Gemma 3
2-way	.50	.46	.49
3-way	.39	.49	.51

Table 3: QWK results for LLMs on trial data.

3.3.1 Closed Source Models

We evaluated gpt-5-mini via the OpenAI API with default model parameters, using task-specific prompts for the 2-way and 3-way settings (see Figure 1 and 2 in the Appendix, respectively). Table 3 shows that gpt-5-mini achieves a QWK of .50 on the 2-way task and .39 on the 3-way task. Performance was measured on the trial data. The 2-way task is comparable to the weaker instance-based models, yet drops substantially for the 3-way task.

3.3.2 Open Source Models

For the open-source models, the same prompt was used for both 2-way and 3-way scoring, since the 2-way rubric subsumes the *partially correct* category under *incorrect*. For the full prompt, see Figure 3 in the Appendix. If a model failed to produce a valid response (i.e. one of the labels), it was re-prompted. If it still failed to answer after it had been prompted 10 times, the prediction was inserted as *invalid*. For evaluation, we took a permissive approach and treated answers with an *invalid* prediction as *correct*.

The two open models we prompted to score answers were Phi-4 (Abdin et al., 2024) and Gemma 3 (27b) (Team, 2025), as these were the two models that performed best in preliminary tests. Both models were accessed via the Ollama API on a local server with default parameters. Table 3 shows model performance on the trial data. Gemma 3 performs better than Phi-4 on both the 2-way and 3-way data. It is just .01 behind GPT5-mini for the 2-way data and outperforms it by .12 QWK on the 3-way data.

For submission on the final test data, we decided on the two open source models, since especially Gemma 3 performed comparably well in both the 2-way and the 3-way setting.

3.4 Influence of Normalization

As part of the similarity-based and instance-based approaches, we also conducted a set of experiments where we used normalized versions of the student answers rather than the raw versions. For normalization, we used TransGEC (Fang et al., 2023) and

MRNH/mbart-german-grammar-corrector², two grammatical error correction systems trained on the Falko-MERLIN dataset for German (Boyd, 2018). In the similarity-based setting, the TransGEC normalization was able to outperform a model using pretrained SBERT on the raw version of the trial data both in the 2-way and 3-way settings. However, when SBERT was finetuned, then normalization proved to have no benefit. In the instance-based approach, the use of normalization mostly proved deleterious, sometimes notably so. Normalization was thus not used in any of the officially submitted runs.

3.5 Classifier Ensembles

Ensemble methods often improve robustness by combining complementary model predictions. This is particularly relevant in our setting, where the individual models differ substantially in how they arrive at their decisions: similarity-based models compare answers to previously seen responses, instance-based models learn direct label mappings, and LLMs rely on rubric-guided prompting.

We therefore evaluated majority-vote ensembles across different classifier combinations. To combine predictions in the 3-way setting, we map labels to numeric values (*incorrect* = 0.0, *partially correct* = 0.5, *correct* = 1.0), average the predictions across models, and round to the nearest valid label.

We evaluated all candidate classifier combinations on the trial data and selected the best-performing ensembles for the official submissions. Overall, the best ensembles consistently outperformed the strongest individual models, indicating that the approaches provide complementary information. Our best ensemble, consisting of *LaSiLearn (mpnet)*, *LaSiLearn_{pre} (distil)*, *SetFit (distil)*, *G-SciEdBERT*, *Phi-4*, and *GPT5-mini* for the 3-way task and *LaSiLearn (mini-lm)*, *SetFit (mpnet)*, and *GSciEdBert* for the 2-way task yielded an improvement from 0.710 on the best individual model to 0.783 QWK on the trial data in the 3-way task and from 0.654 to 0.679 on the 2-way task .

4 Results

Tables 4 (unseen answers) and 5 (unseen questions) present results on the test data of the shared Task. We report Quadratically Weighted Kappa (QWK).

²<https://huggingface.co/MRNH/mbart-german-grammar-corrector>

Model	2-way	3-way
G-SciEdBERT	.59	.64
LaSiLearn	.62	.71
SetFit	.64	.71
best-1	.65	.74
best-2	.66	.76
best-3	.64	.74

Table 4: Evaluation Results for Unseen Answers. Note that best-1 to best-3 are the best-performing classifier combinations on the trial data and are different between 2-way and 3-way.

Unseen Answers Much like on the trial data, instance-based training of G-SciEdBERT is outperformed by the similarity-based models. Between the two similarity-based models, SetFit is again slightly ahead of LaSiLearn on the 2-way data, while both models perform on par on the 3-way data. Our ensemble strategy is able to push performance for both the 2-way and 3-way setting, to .66 and .76, respectively. We submitted the five best-performing runs on the trial data and report the results for the first three of them in Table 4. For details on which models are part of these ensembles, see Table 6 in the appendix. On the test data, the second best run of these five yielded the best performance on the evaluation data, hinting at a certain overfitting in our model selection process. The best-performing submission (best-2) ranked 5th (2-way) and 4th (3-way) in the shared task.

Unseen Questions In addition to the LLMs, we also used the instance-based G-SciEdBERT for prediction on the unseen questions. It falls slightly behind LLM performance, but only by at most .03 QWK on the 2-way data and .02 QWK on the 3-way data. In both settings, Gemma 3 performs best, achieving QWK .39 on 2-way data and QWK .52 on the 3-way data. We also tested voting combinations with three out of the four models (G-SciEdBERT and the three LLMs). These results are included as *no-** in Table 5, indicating which of the four models was *not* part of the voting. In general, voting again increases performance. Curiously, dropping GPT5-mini leads to the lowest increase in performance on 2-way data, but the best result of QWK .58 on the 3-way data. This is likely because gpt already performed poorly on the 3way trial data, but we dont have the numbers for test data. The best result on the 2-way data (QWK .46)

Model	2-way	3-way
G-SciEdBERT	.36	.50
Phi-4	.38	.51
Gemma 3	.39	.52
no_gpt	.42	.58
no_instance	.45	.52
no_gemma	.45	.54
no_phi	.46	.54

Table 5: Evaluation Results for Unseen Questions

is obtained when Phi-4 is not included in the voting. With respect to performance relative to the other submissions to the shared task, we placed 6th in the 2-way condition (no_phi) and 4th in the 3-way condition (no_gpt).

Notably, all models performed significantly better on the 3-way task than the 2-way task. This may be explained by the label distribution: in the 3-way task, the classes are relatively balanced (29.2% Correct, 36.3% Partially Correct, and 34.5% Incorrect). However, in the 2-way task, merging the 'partially correct' and 'incorrect' categories creates a significant class imbalance, with the 'incorrect' category accounting for 70.8% of the data. Separating "partially correct" as its own label in the 3-way task appears to provide a clearer signal for the models to learn.

5 Error Analysis: Influence of 'incoherent' scoring rubrics

A recurring source of confusion during manual inspection of the data was the missing task context: The shared task data originates from a realistic ITS context, but were provided without full context information about the surrounding material. Therefore, a number of questions were not fully transparent such as a question simply stating "Erkläre, in eigenen Worten, warum du dich für diese Lampe entschieden hast." (*Explain in your own words why you have chosen that specific lamp!*), with a rubric for a correct answer stating "Die SuS formulieren eine Begründung basierend auf physikalischen Prinzipien und gesellschaftlichen/ökologischen Beweggründen." (*The students formulate a justification based on physical principles and societal / ecological reasons.*). Without any context it can be hard for humans to determine whether an answer to that question is actually correct.

In an error analysis, we manually annotated such incomplete questions (16 in total) and then compared model performance on these incomplete questions to performance on the remaining trial set. However, QWK values did not substantially differ, hinting at the models capability to infer relevant information from either sample answers or rubric contexts.

6 Conclusion

In this paper, we compared various different approaches to short-answer scoring on a German science question dataset. We found that established instance- and similarity-based methods outperform approaches based on generative LLMs that take only the scoring rubric into account in the unseen answers track, whereas LLM-based approaches where the only applicable option in the unseen questions track. In both cases, we benefit from using an ensemble of classifiers instead of a single classifier only.

Acknowledgements

This work was conducted as part of the DFG-funded ASLAN project (Fördernummer: 563947383).

Limitations

This shared task treated classifier performance as the sole evaluation metric. In a practical application of Short Answer Scoring, however, considerations of robustness and fairness (Loukina et al., 2019; Schaller et al., 2024) can be equally important, particularly with respect to potential biases affecting specific subgroups of learners and should thus also be considered in evaluation.

Ethical Considerations

This work relies on a dataset provided by the shared task organizers. As such, data collection, anonymization, and privacy protection measures were handled prior to our use of the data.

References

Marah Abidin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Xiaoyu Bai and Manfred Stede. 2023. A survey of current machine learning approaches to student free-text evaluation for intelligent tutoring. *International Journal of Artificial Intelligence in Education*, 33(4):992–1030.

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.

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.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2024. Strengths and weaknesses of automated scoring of free-text student answers. *Informatik Spektrum*, 47(3):78–86.

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.

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.

Tao Fang, Xuebo Liu, Derek F. Wong, Runzhe Zhan, Liang Ding, Lidia S. Chao, Dacheng Tao, and Min Zhang. 2023. TransGEC: Improving grammatical error correction with translationese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3614–3633, Toronto, Canada. Association for Computational Linguistics.

Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berrit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, and Hendrik Drachsler. 2026. Report on the bea 2026 shared task on rubric-based short answer scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.

Andrea Horbach and Torsten Zesch. 2019. The Influence of Variance in Learner Answers on Automatic Content Scoring. *Frontiers in Education*, 4.

Ehsan Latif, Gyeong-Geon Lee, Knut Neumann, Tamara Kastorff, and Xiaoming Zhai. 2024. G-sciedbert: A contextualized llm for science assessment tasks in german. *arXiv preprint arXiv:2402.06584*.

- 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.
- 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.
- 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.
- Nils-Jonathan Schaller, Yuning Ding, Andrea Horbach, Jennifer Meyer, and Thorben Jansen. 2024. Fairness in automated essay scoring: A comparative analysis of algorithms on german learner essays from secondary education. In *Proceedings of the 19th workshop on innovative use of nlp for building educational applications (bea 2024)*, pages 210–221.
- Gemma Team. 2025. [Gemma 3 technical report](#).
- Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. [Efficient few-shot learning without prompts](#).

A Appendix

As supplementary material, we include some information that is intended to increase the transparency of our results.

LLM Prompts In Figures 1 (GPT-5 mini, 2-way scoring), 2 (GPT5-mini, 3-way scoring), and 3 (open source LLMs), we include the prompts that were used to score answers with LLMs.

Ensemble Voting In Table 6, we detail which models were part of our best voted scoring runs on the trial data.

Sie sind ein Experte fuer paedagogische Diagnostik im Bereich der deutschen MINT-Bildung. Ihre Aufgabe ist es, Schuelerantworten basierend auf einer vorgegebenen Rubrik zu bewerten.

Sie erhalten:

1. Eine Frage (auf Deutsch)
2. Eine Schuelerantwort (auf Deutsch)
3. Eine Bewertungsrubrik mit Kriterien fuer "Richtig" (Correct) und "Falsch" (Incorrect)

Ihre Aufgabe:

- Lesen Sie die Frage sorgfaeltig durch und verstehen Sie die Aufgabenstellung.
- Analysieren Sie die Antwort des Schuelers.
- Wenden Sie die Rubrik-Kriterien strikt an, um zu entscheiden, ob die Antwort "Correct" oder "Incorrect" ist.
- Beruecksichtigung von Teilverstaendnis: Wenn die Rubrik "Correct" als umfassendes oder vollstaendiges Verstaendnis definiert, muss eine Antwort, die nur Teilaspekte abdeckt, als "Incorrect" bewertet werden.

WICHTIG:

- Antworten Sie NUR mit dem Label: entweder "Correct" oder "Incorrect".
- Geben Sie keine Erklaerungen, Begrueendungen oder zusaetzlichen Text aus.
- Die Bewertung muss sich strikt nach den Kriterien der Rubrik richten.

Figure 1: Prompt for 2-way classification with GPT5-mini.

Configuration	2-way Mix	3-way Mix
best-1	<i>LaSiLearn (mini-lm), SetFit (mpnet), GSciEdBert</i>	<i>LaSiLearn (mpnet) LaSiLearn_{pre} (distil), SetFit (distil), G-SciEdBERT, Phi-4, GPT5-mini</i>
best-2	<i>LaSiLearn (mini-lm), LaSiLearn_{pre} (distil), SetFit (mpnet), Gemma 3, GPT5-mini</i>	<i>LaSiLearn (mini-lm), LaSiLearn (mpnet), LaSiLearn_{pre} (distil), SetFit (distil), G-SciEdBert, RF, Phi-4, GPT5-mini</i>
best-3	<i>LaSiLearn_{pre} (mpnet), SetFit (mpnet), G-SciEdBERT</i>	<i>LaSiLearn (distil), LaSiLearn (mpnet), LaSiLearn_{pre} (distil), G-SciEdBERT, Phi-4, GPT5-mini</i>

Table 6: Details on the best configurations for voting on the trial data.

Sie sind ein Experte fuer paedagogische Diagnostik im Bereich der deutschen MINT-Bildung. Ihre Aufgabe ist es, Schuelerantworten basierend auf einer vorgegebenen Rubrik zu bewerten.

Sie erhalten:

1. Eine Frage (auf Deutsch)
2. Eine Schuelerantwort (auf Deutsch)
3. Eine Bewertungsrubrik mit Kriterien fuer "Richtig" (Correct), "Teilweise richtig" (Partially Correct) und "Falsch" (Incorrect)

Ihre Aufgabe:

- Lesen Sie die Frage sorgfaeltig durch und verstehen Sie die Aufgabenstellung.
- Analysieren Sie die Antwort des Schuelers.
- Wenden Sie die Rubrik-Kriterien strikt an, um zu entscheiden, ob die Antwort "Correct", "Partially Correct" oder "Incorrect" ist.
- Unterscheidung der Stufen: Vergleichen Sie die Antwort sorgfaeltig mit allen drei Rubrik-Stufen. Eine Antwort ist nur "Correct", wenn sie die Kriterien fuer vollstaendige Richtigkeit erfuehlt. Deckt sie nur Teilaspekte ab, ist sie als "Partially Correct" zu bewerten. Fehlt jeglicher inhaltlicher Bezug zu den Rubrik-Kriterien, ist sie "Incorrect".

WICHTIG:

- Antworten Sie NUR mit einem der folgenden Labels: "Correct", "Partially Correct" oder "Incorrect".
- Geben Sie keine Erklaerungen, Begrueendungen oder zusaetzlichen Text aus.
- Die Bewertung muss sich strikt nach den Kriterien der Rubrik richten.

Figure 2: Prompt for 3-way classification with GPT5-mini.

You are a teacher grading student answers for the following question:

Question: {question}

Answers are evaluated with the following rubric:

- {rubric.correct} (correct)
- {rubric.partial} (partially correct)
- {rubric.incorrect} (incorrect)

Please state which label you are assigning the following answer, based on the rubric:

Answer: {answer}

Answer only with the label you are assigning to the answer, without any additional explanation.

Figure 3: Prompting strategy used for evaluation of answers with open-source LLMs.

WSE Research at BEA 2026 Shared Task 2: Multi-Strategy Rubric-Based Short Answer Scoring for German*

Jonas Gwozdz and Andreas Both

Leipzig University of Applied Sciences, Leipzig (Germany)

 Web & Software Engineering Research Group (WSE Research)
{jonas.gwozdz, andreas.both}@htwk-leipzig.de

Abstract

We describe the WSE Research system for the BEA 2026 Shared Task 2 on Rubric-based Short Answer Scoring for German. Our system combines rubric-conditioned prompting with TF-IDF exemplar retrieval, LoRA fine-tuning of open-source Qwen models, and prediction aggregation across complementary scorers. The central question is when prompt engineering, parameter-efficient adaptation, and aggregation each help for rubric-based grading. On the ALICE-LP-1.0 trial set, a fine-tuned Qwen2.5-32B reaches QWK 0.769, surpassing the strongest prompted commercial baseline (Gemini 3 Flash, 0.748). On the official test set, the system ranks second on three tracks and third on the remaining one. Overall, the results show that rubric-conditioned fine-tuning is a competitive and cost-effective alternative to commercial APIs for German short answer scoring, while aggregation helps on seen questions but larger single models generalize better to unseen rubrics.

1 Introduction

Automated short answer grading (ASAG) can reduce assessment effort while maintaining consistent scoring quality (Burrows et al., 2015), but most resources target English. For German, only four dedicated ASAG corpora exist, with approximately 19,500 labeled answers compared to over 44,500 in English (Padó et al., 2024); morphology, compounding, case, and free word order further complicate transfer from English benchmarks (Horbach et al., 2024).

The BEA 2026 Shared Task 2 on Rubric-based Short Answer Scoring for German (Gombert et al., 2026) addresses this gap with the ALICE-LP-1.0 benchmark: 7,899 labeled student answers

across 78 STEM questions, each paired with textual rubrics for **Correct**, **Partially correct**, and **Incorrect**. Because LLM grading accuracy degrades with rubric granularity (Deng et al., 2026) and ASAG transfer across datasets loses 13–47% QWK (Funayama et al., 2023), the task requires per-task adaptation rather than generic grading.

We present the WSE Research system (Figure 1), which combines three complementary strategies that address the following research questions:

- RQ1** To what extent can rubric-conditioned prompt engineering close the performance gap between open-source and commercial LLMs for German short answer scoring?
- RQ2** How does LoRA fine-tuning performance scale with model size for rubric-based short answer classification?
- RQ3** Can ensemble aggregation over heterogeneous scoring strategies outperform individual models?

We evaluate three complementary design choices: rubric-conditioned prompting, LoRA fine-tuning, and aggregation across scorers with different error profiles. The strongest fine-tuned model surpasses the strongest prompted commercial baseline on the trial set (Ferreira Mello et al., 2025); on the official test set, the system places second on three tracks and third on the remaining one. Code and reproducibility artifacts are publicly available.¹

2 Related Work

Automated Short Answer Grading. ASAG has progressed from keyword matching through supervised classifiers to neural approaches (Burrows et al., 2015); the SemEval-2013 winner already combined lexical, syntactic, and character n -gram features via stacking (Heilman and Mad-

*This work was partially funded by the German Federal Ministry of Education and Research (BMBF) under grant FKZ 03FHP239.

¹<https://github.com/WSE-research/bea2026-german-asag>

nani, 2013). A recent survey reports that rubric-guided prompting and human-in-the-loop systems outperform zero-shot single-LLM setups (Nkoyo et al., 2025).

German ASAG. Padó et al. (2024) show that multilingual sentence embeddings are competitive on German but drop 15–20 F1 points on unseen questions, identifying cross-question generalization as the key bottleneck. Horbach et al. (2024) further show that multilingual transformers outperform translate-then-score pipelines across languages, motivating our German-language prompts and native fine-tuning.

LLM-Based Grading and Example Selection. LLMs support rubric-based scoring through in-context learning (Frohn et al., 2025), with rubric-conditioned prompts outperforming rubric-free approaches (Wang et al., 2019); yet fine-tuned smaller models can still outperform vanilla GPT-4 when labeled data is available (Ferreira Mello et al., 2025). Few-shot selection is similarly important: RAG-based retrieval outperforms random selection ($p < 0.001$) (Zhao et al., 2025), and boundary-focused exemplars reduce adjacent-score errors from 0.26 to 0.08 (Chu et al., 2026a). Our TF-IDF-based selection (Section 3.1) combines both signals. Prompt format is known to be brittle: rubric order and score notation can shift grades by up to 45% for frontier models (Deng et al., 2026); automated rubric refinement (Chu et al., 2026b) addresses this at the rubric side but was not attempted here.

Fine-Tuning and Calibration for ASAG. LoRA (Hu et al., 2022) enables parameter-efficient fine-tuning of large models on task-specific data. Funayama et al. (2023) report that rubric-keyphrase fine-tuning improves QWK by approximately 0.25 when training data is scarce. Raikote et al. (2026) reach $\text{QWK} \geq 0.80$ with fine-tuned Qwen-7B on English short answers, while Bexte et al. (2024) show that confidence thresholds are prompt-dependent—a finding our experiments corroborate (Section 5.3). Our work extends LoRA fine-tuning to German with a systematic 7B–72B scaling study.

Taken together, prior work motivates our three-phase design. *Prompt engineering* operationalizes rubric-conditioned in-context scoring, *LoRA fine-tuning* tests whether task-specific supervision can replace commercial prompting, and a *retrieval baseline* anchors both against a non-neural alternative. We leave automated rubric refinement out of

scope because the shared task evaluates systems on fixed organizer-provided rubrics; modifying them would confound scorer quality with rubric rewriting quality.

3 Approach

Our approach is designed as a three-phase rubric-scoring pipeline, as illustrated in Figure 1. First, we generate candidate score labels from the same input triple (question, student answer, rubric) with prompted LLMs, fine-tuned LLMs, and a TF-IDF k NN baseline that labels by similar training answers without neural inference. Second, we compare these scorers under trial-set evaluation to isolate the contribution of prompt engineering, model scale, and aggregation. Third, we choose the final prediction rule per deployment setting: aggregation for seen questions and a single large fine-tuned model for unseen questions, where rubric transfer dominates. The remainder of this section instantiates these phases through prompt engineering (Section 3.1), fine-tuning (Section 3.2), and prediction aggregation (Section 3.3).

3.1 Phase 1: Prompt Engineering

The prompt engineering phase instantiates the pipeline with rubric-conditioned in-context scoring. Each request includes the full textual rubric as the system prompt; the model receives the question and student answer and returns one of the three score labels. We evaluate this setup with Gemini 3 Flash and Qwen3.5-27B to compare a strong commercial API against the strongest self-hosted open-source prompting variant from our model-screening runs. This grounds scoring in the provided rubric rather than the model’s parametric knowledge. We deliberately avoid chain-of-thought prompts, as recent evidence suggests CoT can degrade QWK on educational grading tasks despite improving NLG evaluation (Nkoyo et al., 2025).

TF-IDF-Based Example Selection. Rather than using fixed few-shot examples, we dynamically select training examples for each test instance via TF-IDF cosine similarity between the test answer and training answers for the same question. Examples enter the prompt in two roles: *similar examples*, retrieved by TF-IDF cosine similarity to the test answer among same-question training answers, and *boundary examples*, drawn from less-represented score labels to expose the model to

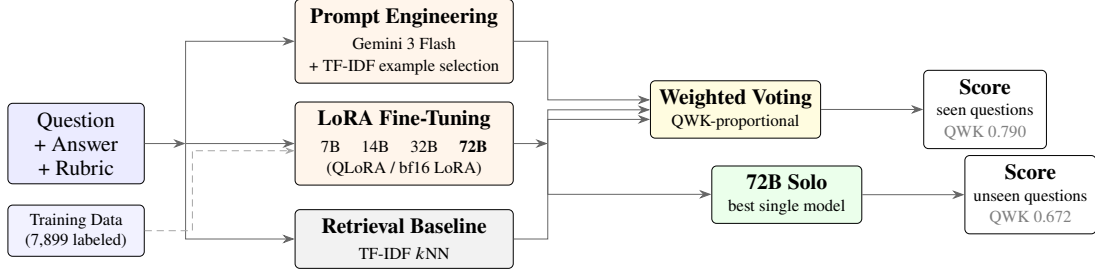


Figure 1: Overview of the WSE Research scoring pipeline. Three phases produce score predictions from the input triple (question, student answer, rubric). For seen questions, weighted voting is best; for unseen questions, the Qwen2.5-72B alone outperforms all ensembles. Dashed arrow: training data flow.

within-rubric edge cases. We quantify question difficulty as the dominant-label fraction in that question’s training subset and bin questions into easy (> 0.60), medium ($0.40\text{--}0.60$), and hard (< 0.40) tiers; pilot sweeps showed two similar examples to be consistently helpful, while the useful number of boundary examples varied with difficulty. The final Gemini configuration, therefore, fixes the similar-example count at two and uses 1, 2, or 3 boundary examples for easy, medium, or hard questions, respectively. This combines retrieval of semantically close answers (Zhao et al., 2025) with boundary-focused calibration (Chu et al., 2026a). We iterated prompts systematically (6 strategies for Gemini 3 Flash, 33 variants across 8 rounds for Qwen3.5-27B); the checkpoint-specific transfer behavior this revealed is analyzed in Section 5.3.

3.2 Phase 2: Fine-Tuning

The fine-tuning phase asks how much rubric-conditioned supervision can be absorbed by open-source models of different sizes. We apply Low-Rank Adaptation (LoRA, cf. (Hu et al., 2022)) to Qwen-family models at four Qwen2.5 scales (Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B, Qwen2.5-72B) plus Qwen3.5-9B to test whether a newer architecture improves fine-tuning efficiency at smaller scale. The Qwen3.5-27B model in Section 3.1 is used only for prompting, not for LoRA training.

Instruction Format. Each training example is formatted as an instruction-following turn: rubric text as system prompt, question and student answer as user message, gold score label as assistant response. This aligns fine-tuning with the inference-time interface from the prompt-engineering phase, so differences between prompting and fine-tuning can be attributed mainly to parameter adaptation rather than interface changes. Following Funayama et al. (2023), we include rubric key phrases

as explicit input features.

Training Configuration. Preliminary studies using Qwen2.5-7B indicated that one epoch underfit and five epochs overfit, while increasing the LoRA rank beyond 32 did not improve trial performance. We therefore use a single pragmatic configuration across scales ($r = 32$, $\alpha = 32$, learning rate 2×10^{-4} , 3 epochs) to isolate model-size effects rather than retuning each model separately. The 14B, 32B, and 9B models use full bf16 LoRA on L40S GPUs; the Qwen2.5-7B and Qwen2.5-72B use 4-bit QLoRA (NF4), with the 72B moved to an H200 to fit memory constraints. We train each size in two configurations: *train-only* (for unbiased trial-set comparison) and *all-data* (train + trial, for final submission). Trial labels are released development data: train-only models provide an unbiased internal comparison, whereas all-data models use every labeled example available before hidden-test submission.

3.3 Phase 3: Prediction Aggregation

The aggregation phase asks whether complementary scorers should be combined or deployed separately. For unbiased trial-set analysis, we train a logistic-regression meta-learner (Heilman and Madnani, 2013) with leave-one-question-out cross-validation. Component models span three fine-tuned variants (Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B), the best prompt-only result (Qwen3.5-27B), and a non-neural TF-IDF k -nearest neighbor baseline. We set $k = 7$ because it was the strongest pure-retrieval setting in pilot runs, outperforming $k \in \{3, 5, 11\}$. Features comprise one-hot model predictions, k NN confidence, answer length, and question difficulty. For final submissions, we use simpler rules: QWK-weighted voting for seen questions and the 72B solo model for unseen questions, because weaker models diluted the 72B’s rubric generalization.

4 Experimental Setup

Dataset. ALICE-LP-1.0 (Gombert et al., 2026) contains 7,899 German secondary-school STEM answers to 78 questions, each paired with a textual rubric defining three score levels. Train (7,072) and trial (827) sets are stratified by question (29% **Correct**, 36% **Partially correct**, 35% **Incorrect**). The test set (5,094 samples) splits into 2,008 answers to the 78 seen questions and 3,086 answers to 39 entirely new questions with 34 new rubrics. Each split is graded under both a 3-way label scheme and a 2-way scheme (with **Partially correct** collapsed into **Incorrect**), yielding the four official shared-task tracks.

Models and Hardware. We evaluate two prompt-only baselines (Gemini 3 Flash, parameter count undisclosed; Qwen3.5-27B) and five fine-tuned Qwen variants (Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B, Qwen2.5-72B, Qwen3.5-9B), all trained with LoRA rank $r = 32$ for 3 epochs. Experiments up to 32B run on 2xL40S GPUs using vLLM (Kwon et al., 2023); the Qwen2.5-72B is fine-tuned on a single H200 NVL GPU via PEFT (Mangrulkar et al., 2023).

Evaluation. The primary metric is Quadratic Weighted Kappa (QWK), the official shared-task ranking metric; we additionally report accuracy. Trial-set evaluations use train-only models for unbiased comparison; submission models use all available data (train + trial).

5 Results

5.1 Trial Set Results

Table 1 reports trial-set performance with train-only fine-tuned models.

Fine-tuning consistently outperforms prompt engineering: the Qwen2.5-14B already surpasses Gemini 3 Flash (0.005 QWK), and the Qwen2.5-32B extends the margin to 0.021. A QWK-proportional weighted vote over the four-model pool {Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B, Gemini 3 Flash}—with weights derived from each model’s leave-one-question-out cross-validated QWK on the training set and normalized to sum to one—achieves the highest unbiased trial QWK of 0.781 by combining fine-tuned and prompt-based models; a logistic-regression stacker over an extended five-candidate pool that additionally includes the TF-IDF k NN baseline reaches 0.776 under leave-one-question-out cross-validation. This

Phase	System	QWK	Acc
<i>Prompt Engineering</i>	Gemini 3 Flash	0.748	73.6
	Qwen3.5-27B	0.719	70.2
<i>Fine-Tuning (train-only $\rightarrow$ trial)</i>			
	Qwen2.5-7B QLoRA	0.726	70.9
	Qwen2.5-14B LoRA	0.753	74.1
	Qwen2.5-32B LoRA	0.769	75.7
	Qwen2.5-72B QLoRA	0.768	75.7
	Qwen3.5-9B LoRA	0.756	74.2
<i>Ensemble / Baseline</i>			
	Weighted vote (4 models)	0.781	— [†]
	LogReg stacking (5-cand.)	0.776	75.5
	TF-IDF k NN ($k = 7$)	0.612	64.5

Table 1: Results on the ALICE-LP-1.0 trial set (827 samples). Fine-tuned models use train-only data for unbiased evaluation. [†] Accuracy is omitted for the QWK-proportional weighted vote because the aggregation produces a continuous score in $[0, 2]$ rather than a discrete label, and any threshold-based discretization is a separate decision rule unrelated to the QWK ranking metric.

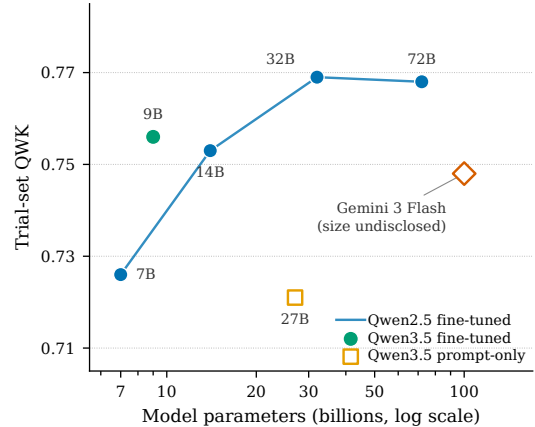


Figure 2: Trial-set QWK versus model size: fine-tuned Qwen variants (filled) saturate by 32B; prompt-only baselines (hollow); Gemini 3 Flash is plotted at a nominal 100B (true size undisclosed).

answers **RQ1**: rubric-conditioned prompt engineering yields a strong baseline, but it does not close the gap to fine-tuned open-source models once task-specific supervision is available.

Scaling Behavior. Figure 2 visualizes trial-set QWK as a function of model size. Fine-tuning QWK scales with diminishing returns and saturates by 32B (−0.001 from 32B to 72B). The 7B (0.726) already matches the prompt-only Qwen3.5-27B (0.719); Qwen3.5-9B (0.756) outperforms Qwen2.5-14B (0.753), suggesting that model generation matters more than parameter count. This answers **RQ2**: larger fine-tuned models generalize more robustly across rubrics, but

Track	Ours	IWM-DKM	Rank
<i>3-way scoring</i>			
Seen questions	0.790	0.796	2nd
Unseen questions	0.672	0.681	2nd
<i>2-way scoring</i>			
Seen questions	0.717	0.726	2nd
Unseen questions	0.533	0.550	3rd

Table 2: Test-set QWK on all four official tracks for our system and the winning team, IWM-DKM. Our submitted systems are the QWK-weighted vote of {Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B, Gemini 3 Flash} on 3-way seen, the binary majority vote of {Qwen2.5-14B, Qwen2.5-32B, Gemini 3 Flash} on 2-way seen, and the Qwen2.5-72B solo on both unseen-question tracks. Retraining the 72B on train + trial lifts its trial QWK from 0.768 (train-only, Table 1) to 0.851 (in-sample), a 0.083 headroom from incorporating trial labels at submission time.

unbiased trial performance saturates by 32B.

5.2 Test Set Results

Table 2 reports our best system on each of the four official tracks alongside the corresponding result of the winning team IWM-DKM.

On *seen questions*, the weighted ensemble (QWK 0.790) outperforms all individual models, mirroring the trial-set pattern. On *unseen questions*, the Qwen2.5-72B alone (QWK 0.672) beats a 72B-inclusive weighted vote (0.653) and the Qwen2.5-32B solo (0.611); adding weaker models *hurts*, and the seen-to-unseen drop is 0.118 QWK for 72B versus 0.169 for 32B. This answers RQ3: aggregation helps on seen questions, but becomes harmful on unseen ones once a single strong model dominates.

5.3 Analysis

Prompt-Model Coupling and Calibration. Optimal prompts are checkpoint-specific: prompts tuned for Gemini 3 Flash degrade by 0.260–0.400 QWK on other Gemini variants, aligning with Deng et al. (2026) on format sensitivity. Fine-tuned models exhibit >99% logprob-confidence, making confidence-based routing infeasible without post-hoc calibration (Raikote et al., 2026).

Error Patterns. On the trial set (32B train-only), 94.3% of misclassifications are adjacent-boundary errors, and **Partially correct** has the lowest per-label precision (65.0%), consistent with rubric-granularity findings (Deng et al., 2026). Cross-model agreement drops from 53.9% on seen to 41.4% on unseen, where the Qwen2.5-72B most often corrects majority leniency by downgrading.

6 Conclusion

We presented the WSE Research system for the BEA 2026 Shared Task 2, placing second on three of four tracks. Rubric conditioning matters, parameter-efficient fine-tuning exploits it most reliably and saturates by 32B (RQ1, RQ2), and aggregation helps on seen questions but hurts unseen ones (RQ3). Rubric-conditioned fine-tuning is thus a practical open-source alternative.

Limitations

Our system was developed and evaluated exclusively on the ALICE-LP-1.0 dataset, which covers German STEM answers at the secondary-school level. Generalization to other languages, educational levels, or subject domains has not been tested. Prior work suggests that ASAG strategies degrade by 13–47% QWK when transferred across datasets (Funayama et al., 2023), and cross-corpus transfer for German specifically can incur up to 22 percentage points of accuracy loss (Padó et al., 2024).

The 72B out-of-sample trial QWK (0.768) shows no improvement over the 32B (0.769), suggesting that the scaling curve plateaus at 32B for this dataset size (7,072 training samples). Retraining the 72B on train + trial for submission lifts its trial QWK to 0.851 (in-sample), a 0.083 headroom over the directly comparable train-only value; this gain is observed on data the model has seen and should not be read as out-of-sample improvement.

We rely on the provided rubrics as-is and do not perform rubric quality assessment or optimization. Approximately 10% of questions use vague competency-level descriptions without specific criteria, which may limit scoring accuracy for both our system and human raters. Confusion-aware rubric refinement (Chu et al., 2026b) could address this but was not attempted in the current work.

All evaluations use automated metrics (Quadratic Weighted Kappa, accuracy); we did not conduct human evaluation of model predictions or analyze inter-annotator agreement on the dataset.

Fine-tuned models show near-uniform confidence (>99%), which prevents meaningful confidence-based quality control. This limits the applicability of human-in-the-loop strategies that rely on uncertainty estimates for selective routing (Bexte et al., 2024). Post-hoc temperature scaling (Raikote et al., 2026) may restore useful confidence

signals but was not applied here.

Ethics Statement

This work uses the publicly released ALICE-LP-1.0 dataset, which was collected with appropriate institutional oversight within the ALICE project (Leibniz Foundation). Student answers are anonymized and contain no personally identifiable information.

Automated scoring systems should be deployed as decision-support tools rather than sole arbiters of student grades. We note that LLM-based scoring can reproduce or amplify biases present in training data, including potential biases in rubric design or human scoring patterns. Deployment in high-stakes assessment contexts requires human oversight and ongoing bias monitoring.

Generative AI Disclosure. Parts of the codebase were developed with the assistance of AI agents. All experimental design, analysis, and writing were conducted by the authors.

References

- Marie Bexte, Andrea Horbach, Lena Schützler, Oliver Christ, and Torsten Zesch. 2024. [Scoring with confidence? – Exploring high-confidence scoring for saving manual grading effort](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 119–124. Association for Computational Linguistics.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. [The eras and trends of automatic short answer grading](#). *International Journal of Artificial Intelligence in Education*, 25(1):60–117.
- Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Kevin Haudek, Joseph Krajcik, and Jiliang Tang. 2026a. [Optimizing in-context demonstrations for LLM-based automated grading](#). *Preprint*, arXiv:2603.00465.
- Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Joseph Krajcik, Namsoo Shin, and Jiliang Tang. 2026b. [Confusion-aware rubric optimization for LLM-based automated grading](#). *Preprint*, arXiv:2603.00451.
- Haotian Deng, Chris Farber, Jiyeon Lee, and David Tang. 2026. [Rubric-conditioned LLM grading: Alignment, uncertainty, and robustness](#). *Preprint*, arXiv:2601.08843.
- Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. 2025. [Automatic short answer grading in the LLM era: Does GPT-4 with prompt engineering beat traditional models?](#) In *Proceedings of the 15th International Learning Analytics and Knowledge Conference (LAK 2025)*, pages 93–103.
- Scott Frohn, Tyler J. Burleigh, and Jing Chen. 2025. [Automated scoring of short answer questions with large language models: Impacts of model, item, and rubric design](#). In *Artificial Intelligence in Education (AIED 2025)*, volume 15882 of *Lecture Notes in Artificial Intelligence*, pages 44–51. Springer.
- Hiroaki Funayama, Yuichiroh Shindo, Hiroki Ouchi, and Kentaro Inui. 2023. [Reducing the cost: Cross-prompt pre-finetuning for short answer scoring](#). In *Proceedings of the 24th International Conference on Artificial Intelligence in Education (AIED 2023)*.
- Sebastian Gombert, Zhifan Sun, Jannik Lossjew, Tobias Wyrwich, Berrit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, Fabian Zehner, and Hendrik Drachler. 2026. BEA 2026 shared task: Rubric-based short answer scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. To appear. Task page: <https://edutec.science/bea-2026-shared-task/>.
- Michael Heilman and Nitin Madnani. 2013. [ETS: Domain adaptation and stacking for short answer scoring](#). In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279.
- Andrea Horbach, Joey Pehlke, Ronja Laarmann-Quante, and Yuning Ding. 2024. [Crosslingual content scoring in five languages using machine-translation and multilingual transformer models](#). *International Journal of Artificial Intelligence in Education*, 34(4):1294–1320.
- 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 *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP 2023)*, pages 611–626.
- Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2023. [PEFT: Parameter-efficient fine-tuning of billion-scale models on low-resource hardware](#).
- Fredrick Eneye Tania-Amanda Nkoyo, Chukwuebuka Fortunate Ijezue, Maaz Amjad, Ahmad Imam Amjad, Sabur Butt, and Gerardo Castaneda-Garza. 2025. [Advances in auto-grading with large language](#)

models: A cross-disciplinary survey. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*.

Ulrike Padó, Yunus Eryilmaz, and L. Lynette Kirschner. 2024. [Short-answer grading for german: Addressing the challenges](#). *International Journal of Artificial Intelligence in Education*, 34(4):1321–1352.

Pranav Raikote, Korbinian Randl, Ioanna Miliou, Athanasios Lakes, and Panagiotis Papapetrou. 2026. [CHiL\(L\)Grader: Calibrated human-in-the-loop short-answer grading](#). *Preprint*, arXiv:2603.11957.

Tianqi Wang, Naoya Inoue, Hiroki Ouchi, Tomoya Mizumoto, and Kentaro Inui. 2019. [Inject rubrics into short answer grading system](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 175–182.

Chenyang Zhao, Mariana Silva, and Seth Poulsen. 2025. [Language models are few-shot graders](#). In *Artificial Intelligence in Education*, Lecture Notes in Computer Science, pages 3–16. Springer Nature Switzerland.

A Prompt Templates

This appendix gives the verbatim wording of the two prompt families used in the prompt-engineering phase (Section 3.1) and as the inference-time interface for the fine-tuning phase (Section 3.2).

Prompted scoring (Gemini 3 Flash, Strategy C5c). The user prompt contains the question, the verbatim three-level rubric, two retrieved similar examples, $n_b \in \{1, 2, 3\}$ boundary examples (selected as described in Section 3.1), and finally the test answer. The system prompt is verbatim:

Du bist ein automatisches Bewertungssystem für Schülerantworten in MINT-Fächern.

AUFGABE

Bewerte die Antwort eines Schülers anhand der Bewertungsrubrik und der Beispielerwertungen.

BEWERTUNGSSTUFEN

- ‘Correct’: Die Antwort trifft ALLE zentralen Punkte der Correct-Rubrik.
- ‘Partially correct’: Die Antwort trifft MINDESTENS EINEN zentralen fachlichen Punkt der Rubrik korrekt, verfehlt aber andere wesentliche Kriterien.
- ‘Incorrect’: Die Antwort trifft KEINEN der fachlichen Kernpunkte der Rubrik.

ENTSCHEIDUNGSREGELN

1. Entscheide anhand der Rubrik UND konsistent mit den Beispielerwertungen.

2. Bewerte nur, was geschrieben wurde --- keine wohlwollenden Annahmen.

3. GRENZE Incorrect vs. Partially correct:

- Vage Aussagen, Alltagswissen oder Umformulierungen der Frage OHNE fachlichen Inhalt aus der Rubrik sind IMMER Incorrect.

- Partially correct erfordert NACHWEISBAR mindestens einen konkreten fachlichen Punkt, der in der Rubrik als Kriterium genannt wird.

4. GRENZE Partially correct vs. Correct:

- Correct erfordert, dass die Antwort die Rubrik-Kriterien VOLLSTÄNDIG abdeckt.

- Wenn die wesentlichen Konzepte korrekt und vollständig dargelegt sind, kleine sprachliche Ungenauigkeiten aber vorliegen -> trotzdem Correct.

- Im Zweifel: orientiere dich an den Beispielerwertungen für diese Frage.

5. Leere Antworten, Nichtwissen (‘?’ , ‘Keine Ahnung’), einzelne Wörter ohne Erklärung -> IMMER Incorrect.

FORMAT

Antworte NUR mit einem JSON-Objekt:

```
{‘score’: ‘Correct’
| ‘Partially correct’ |
‘Incorrect’, ‘confidence’:
0.0--1.0}
```

Fine-tuned scoring (Qwen2.5 LoRA). The fine-tuned models receive a minimal instruction-format prompt that mirrors the inference contract and dispenses with explicit decision rules, since those have been internalized via training:

System: Du bist ein Bewertungssystem für Schülerantworten. Bewerte die Antwort anhand der Rubrik. Antworte ausschliesslich mit JSON: {‘score’: ‘Correct’ | ‘Partially correct’ | ‘Incorrect’}

User: Frage: {question} \n
 Bewertungsrubrik: - Correct: {rubric.Correct} - Partially correct: {rubric.PC} - Incorrect: {rubric.Inc} \n
 Schülerantwort: {answer}

The full executable prompt code, including example selection and all model-specific variants, is publicly available in our repository (see Section 1).

AMATI at BEA 2026 Shared Task 2: Automatic Short Answer Grading with Inductive Logic Programming and a Large Language Model

Alistair Willis

School of Computing and Communications
The Open University
Milton Keynes, UK
alistair.willis@open.ac.uk

Aisling Third

Knowledge Media Institute
The Open University
Milton Keynes, UK
aisling.third@open.ac.uk

Abstract

We discuss the AMATI submission to the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German. Our neuro-symbolic system uses a combination of symbolic rules, automatically learned with a form of Inductive Logic Programming, and the Mistral-large language model. We wanted to investigate whether the combination would improve overall grading performance, while using the automatically induced symbolic rules for explainability, and the LLM for robustness. We find that the combination of approaches resulted in improved overall performance for the 3-way task. However, including the symbolic rules did not improve upon Mistral’s performance in the 2-way test.

This paper presents our approach to the unseen answers challenges. Our team finished 6th out of 9 in the 2-way challenge, and 5th out of 8 in the 3-way challenge. In the 3-way challenge, neither our symbolic system nor the use of Mistral alone would have placed higher than 6th of the 8 competitors, illustrating the improvement of the combined approach over either of the individual approaches.

1 Introduction

Assessment is a core aspect of education. Formative assessment in particular is a valuable aid to student’s self learning (Clark, 2012). Techniques such as Automatic Short Answer Grading (ASAG) (“grading” and “scoring” both appear in the literature) can potentially support scalable student learning by providing low cost feedback on low-stakes assessment (Aggarwal et al., 2025).

Recent work in ASAG has been dominated by approaches based on statistical techniques or deep learning/large language models (Lu et al., 2025). In this work, grading is typically treated as a classification task, in which student responses are assigned to one of a number of potential grades. This BEA task (Gombert et al., 2026) follows similar tasks

such as the SemEval 2013 Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge (Dzikovska et al., 2013) in providing a 2 way task (student responses are classified as correct or incorrect) and a 3 way task (in which responses are classified as correct, partially correct or incorrect).

Large language models (LLMs) such as Mistral (Mistral AI, 2024) typically achieve strong empirical performance, but the resulting models are not usually interpretable by humans. Symbolic learning techniques, such as Inductive Logic Programming (ILP) (Quinlan, 1990; Muggleton and Feng, 1990) aim to induce models as logic programs rather than weighted networks. Such logic programs are generally more interpretable by humans, as well as being editable if desired. For an ILP approach to ASAG, the goal is to induce a set of rules which will assign grades from a symbolic representation of the question, rubric and student response. When applying the rules to unseen instances, the reasoning behind the allocated grade is clear. This explainability will be crucial for broader adoption of ASAG technology in the classroom.

This approach to using symbolic rules to augment neural language models is becoming of interest as the need for human-understandable explanations is increasingly recognised (Bhuyan et al., 2024; Yang et al., 2025). The neuro-symbolic system AlphaGeometry (Trinh et al., 2024) has similarly been developed to generate human readable explanations of its reasoning (in this case, geometry problems). Dinu et al. (2024) present a more general framework to address the general task of embedding logic solvers into generative models.

Although theories induced by ILP are quite explanatory (Willis, 2015), LLMs have demonstrated their robustness in the face of the noisy and poorly formed language that is typical in short student responses. In this paper, we discuss our attempt to combine the explanatory nature of ILP with the

robustness of an LLM. Our system was applied to the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German (Gombert et al., 2026) for an independently verified evaluation. Our code is available through GitHub¹.

2 Grading rule induction with Answer Set Programming

ILP is an optimisation task: we want to find the logic program which most accurately classifies the training cases. As such, much of the foundational work in ILP focussed on optimising search algorithms (Muggleton and Feng, 1990; Quinlan, 1990; Muggleton, 1995). However, some recent approaches have used Answer Set Programming (ASP) solvers to handle the optimisation (Law et al., 2014; Cropper et al., 2022).

We follow the same strategy, using the ASP solver *clingo* (Gebser et al., 2019) to optimise the rule coverage. This approach then implements an ILP-style rule induction: ASP is repeatedly used to optimise individual rules until a maximal set of training cases is covered.

Our approach consists of three components: a representation of student responses and rubric information as logical facts (section 2.1), a set of templates for classification rules (section 2.2), and a procedure for selecting rules that explain the training data (section 2.3).

2.1 Representing ASAG as a logic program

ILP requires that the information about the task be represented in the same language as the induced rules. As such, the first step is to represent the training data (that is, the questions, rubrics, student responses and ground truth gradings) in a logical format.

For the unseen answers task, each question is treated separately. For each task, we then have:

- a single question (identifier and text),
- a grading rubric (text for each possible grade),
- a set of student responses (each with an identifier and its own text), and
- a ground truth grade (“score”) for each student response.

We refer to each of the texts of each question, rubric and student response as a *document*. Each document was converted to a sequence of lemmas using spaCy (2020) with the `de_core_news_md` model. Each lemma is then encoded as a predicate of the form:

```
lemma_in_doc(Doc, Position, Lemma)
```

where `Doc` is the document’s index, `Position` is the token index, and `Lemma` is the lemmatised form of the token. The ground truth grades are then encoded as:

```
label(ResponseID, Class).
```

where `Class` is one of `{correct, incorrect}` or `{correct, partially_correct, incorrect}` depending on the task.

As an example, the (slightly simplified) encoding of one of the questions from the training data is given in figure 1. In the encoding, lemmas in the question are indexed as `q`, in the rubric as `c`, and in the student responses as `r_01`, `r_02`, and so on. This encoding allows rules to refer directly to linguistic features. For example, a rule can test whether a response contains a particular concept (represented as a lemma), or whether that concept appears in the rubric but not in the question. The representation of the documents’ indices in the predicates also enables rules to express relationships between documents, rather than treating the response in isolation.

2.2 Representing grading rules

Grading rules are expressed as logical clauses that specify the conditions under which a response should be assigned a particular label. Following the logic-based representation of the documents, the grading rules can be clearly expressed in the language of logic programming. For example, the covers predicate in figure 2 states that `rule3` applies to any student response whose lemmatised form contains the lemmas `kurve` and `steigen`. The additional clause `I < J` also allows us to express that for a match, the lemmas must appear in that order (ie. `kurve` precedes `steigen`). The `predicted_grade` states that any response which matches the condition in `covers` is graded as `partially_correct` by the system.

A more complex example is given in figure 3. Here, a response is covered by the rule if the response contains the lemma `reagieren`, the rubric

¹https://github.com/undercertainty/AMATI_BEA_submission

```

lemma_in_doc(q, 0, "welche").
lemma_in_doc(q, 1, "dir").
lemma_in_doc(q, 2, "bereits").

:

lemma_in_doc(c, 0, "die").
lemma_in_doc(c, 1, "schüler").
lemma_in_doc(c, 2, "innen").

:

lemma_in_doc(r_01, 0, "die").
lemma_in_doc(r_01, 1, "steigung").
lemma_in_doc(r_01, 2, "der").

:

lemma_in_doc(r_02, 0, "die").
lemma_in_doc(r_02, 1, "tangente").
lemma_in_doc(r_02, 2, "berührt").

:

grade(r_01, correct).
grade(r_02, incorrect).

```

Figure 1: Encoding of training data. Question id=67c1649e-b694-4813-be49-abfd65a6798e

```

covers(rule3, Response):-
    student_response(Response),
    lemma_in_doc(Response, I, "kurve"),
    lemma_in_doc(Response, J, "steigen"),
    I < J.

predicted_grade(partially_correct).

```

Figure 2: Grading rule with ordered lemmas and predicted grade

also contains reagieren, but reagieren does not appear in the question. In this case, responses which match are graded as correct by the system.

Figure 3 implements Pandey’s (2022) concept of *key information*: new information (that is, information which does not appear in the question) which must appear in the student response. The key information can be identified as it appears in the grading rubric, but not in the question itself.

```

covers(ki_rule1, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, "reagieren"),
    lemma_in_doc(c, _, "reagieren"),
    not lemma_in_doc(q, _, "reagieren").

predicted_grade(correct).

```

Figure 3: Grading rule with key information and predicted grade

2.3 Rule and theory induction via ASP

In order to generate a complete set of rules (a “theory”) to grade all the responses to a given question, we use an ASP solver to generate individual rules. To generate specific rules, we define a set of generalised *rule templates*.

```

covers(ki_rule1, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, Lemma),
    lemma_in_doc(c, _, Lemma),
    not lemma_in_doc(q, _, Lemma).

predicted_grade(Grade).

```

Figure 4: Rule template. Lemma and Grade are found by optimising with the answer set solver

An example template is given in figure 4. The figure represents the template form of the rule shown in figure 3. Then given the encoding of the question, rubric and a set of responses, the best *specific* rule is found from the values of:

- the choice of rule (from all the templates),
- the values of each Lemma variable in the chosen rule, and
- the predicted grade (ie. Grade)

which maximises the correct number of predicted grades from the set of student responses. To find these optimum values, we use the *clingo* answer set grounder and optimiser to find the optimum values for Grade and Lemma. (The optimised value is actually compression (Muggleton, 1995) rather than coverage).

To generate a complete theory from the initial set of student responses, we use a very simple procedure:

1. The theory is initially an empty list.
2. Find the specific rule which correctly predicts the grade for the largest number of student responses.
3. Add the specific rule to the current theory.
4. Remove all student responses which are correctly predicted by the new rule.
5. Repeat stages 2 to 4 until either all responses are covered, or no new rule can be found.

```

covers(rule1, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, Lemma).

covers(rule2, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, Lemma1),
    lemma_in_doc(Response, _, Lemma2).

covers(rule3, Response):-
    student_response(Response),
    lemma_in_doc(Response, I1, Lemma1),
    lemma_in_doc(Response, I2, Lemma2),
    I1 < I2.

covers(ki_rule1, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, Lemma),
    lemma_in_doc(c, _, Lemma),
    not lemma_in_doc(q, _, Lemma).

covers(ki_rule2, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, Lemma1),
    lemma_in_doc(c, _, Lemma1),
    lemma_in_doc(Response, _, Lemma2),
    lemma_in_doc(c, _, Lemma2),
    not lemma_in_doc(q, _, Lemma1),
    not lemma_in_doc(q, _, Lemma2).

covers(ki_rule3, Response):-
    student_response(Response),
    lemma_in_doc(Response, _, Lemma1),
    lemma_in_doc(c, _, Lemma1),
    lemma_in_doc(Response, _, Lemma2),
    lemma_in_doc(c, _, Lemma2),
    not lemma_in_doc(q, _, Lemma1).

```

Figure 5: Complete list of induction rule templates

This procedure then generates a list of rules which can be applied to new examples. The final output of the process is a theory consisting of an ordered set of rules. Each rule defines the coverage criteria for student responses, and the predicted grade for any responses which match those criteria.

To apply the rules to new cases, each rule in the theory is applied in order. The first matching rule is used to predict the grade. If no rule matches, the prediction defaults to incorrect.

The performance of this system without additional input from a LLM is shown in table 1 as “Rule-based”. The complete set of the rule templates is given in figure 5.

2.4 Limitations and motivation for a hybrid approach

Although some linguistic variation in the student responses is accommodated by using spaCy’s lemmatisation and spell checking, the primary limitation of this method is the reliance on lexical matching. The range of possible responses from students can make the symbolic rules quite fragile. We therefore

wished to explore how integrating these rules with a large language model might address this limitation.

3 Combining with a LLM

For the unseen answers task, we identified several potential approaches to combining the ILP rules with a LLM:

Mistral No ILP rules are used. For each test response, the LLM is supplied with the corresponding question and all responses in the training data for the same question, and asked to predict the correct grade.

Test ILP As *Mistral*, but also including the specific ILP prediction for the test question and the rule(s) applied to make it.

Theory As *Mistral*, but also including the complete set of rules in the theory generated by the ILP process for the task.

Test ILP and Theory As *Mistral*, including the ILP prediction and rule(s) for the test question and the complete theory.

For the unseen questions task, there is of course no “corresponding question”; our intent was to select a sample of questions from the training data instead. In practice, however, the results for the unseen questions task without the LLM did not seem to justify the time and inference cost, so we did not run the LLM cases for that task.

We decided to use the *mistral-large* model (Mistral AI, 2024) as the LLM component, via the paid Mistral API. This choice was based on its performance in the leaderboards in (Thakur, 2025) for performance in German language tasks, and its context window (the length of input it can accommodate) of 256k tokens. The prompts are in the form of templates, with placeholders to insert data, and describe the relevant scenario and what is expected from the model, and include descriptions of input and output data JSON formats. The full prompt templates are in the associated code repository. Figure 6 illustrates the prompt templates.

The output included a field for the LLM’s natural language *explanation* of the given grade, to provide data for future analysis of the LLM’s reasoning beyond the scope of the shared task. There are multiple avenues for this analysis, the most immediate of which is to compare LLM reasoning with the ILP rules, and evaluate its quality; further

You are a helpful and precise assistant for marking students' answers. You will be given a question, a student's answer, and the marking rubric for the question{, and a predicted score with explanation based on an inductive logic programming system}. The possible score values are "Incorrect", {"Partially Correct", }and "Correct". Your task is to evaluate each student's answer against the rubric for that question, and provide a score.

The following, between "TRAINING DATA STARTS" and "TRAINING DATA STOPS", is a JSON array of objects ... JSON details omitted ... representing the score assigned to the student's answer. Consider this as training data to improve your performance in the task.

...
{The following is the full theory generated by inductive logic programming over the training data:}

Figure 6: Prompt template (condensed for space). Scenario variations are indicated by braces.

	QWK	WP	WR	WF1
2-way				
Rule-based	0.489	0.801	0.807	0.795
Mistral	0.644	0.857	0.843	0.847
Theory	0.644	0.857	0.843	0.847
3-way				
Rule-based	0.614	0.662	0.655	0.650
Mistral	0.721	0.755	0.711	0.715
Theory	0.749	0.760	0.736	0.739

Table 1: Results for 3-way Unseen Answer tasks, showing Quadratic Weighted Kappa (QWK), and Weighted Precision (WP), Recall (WR), and F1-Score (WF1). QWK was used to rank task entrants.

possibilities include training new ILP rules from the LLM's explanations.

Testing with the provided trial data revealed that *Test ILP* (with or without the full theory) performed worse across all measures, whereas the *Mistral* and *Theory* cases performed similarly to each other, and both better than the ILP-only case.

We therefore decided to use the *Mistral* and *Theory* cases for the full evaluation.

4 Results and discussion

Our results for the 2-way and 3-way unseen answers are shown in table 1. When the ILP rules were included in the Mistral prompt, the 2-way results showed no improvement. However, for the 3-way task, including the symbolic rules in the

prompt raised the overall performance (as measured by QWK) from 72.1% to 74.9%. This also represents a position in the final evaluation table: our final position was fifth of eight.

5 Conclusion

In this paper, we have discussed how the performance of a system for student response grading was improved by incorporating a set of induced symbolic rules into the neural reasoning process. Although the final results are slightly below the average system performance in the task (6th of 9 in the 2-way task and 5th of 8 in the 3-way task), our interest is the combination of human-interpretable rules with LLM robustness. As illustrated in section 2.2, the rules generated from the templating system are human readable (subject to being comfortable with information being presented in a formal language).

In fact, our longer term aim with this work is not (directly) to improve the overall grading performance: after all, our results show that Mistral alone gives relatively good performance. Rather, we aim to improve the explainability of awarded grades for the benefit of educators (who are ultimately responsible for grades awarded) and students (whose self-directed learning will be improved with better feedback). In a practical classroom context, it is more valuable for an AI to be able to explain why it assigned a particular grade, rather than focussing solely on performance figures.

The key question we wish to address in the next stage of this work is: Do the grades awarded by the combined system closely follow the intended interpretation of the symbolic rules? The aim of combining the symbolic rules with a LLM, is for the LLM to handle linguistic variation, while the symbolic rules handle grading of the content. As such, we want the induced rules to be editable by a human, possibly even a human who is not a computing expert, but who might be an educator in the domain of the questions. In these cases, it is vital that the application of the system primarily follows the symbolic rules (which are human editable) rather than the LLM inferences (which generally are not). Indeed, the results of LLM outputs are not always replicable (Cui and Alexander, 2026). Our further analysis will therefore concentrate on understanding the relations between the symbolic rules and the performance of the complete grading system.

Limitations

Our method makes use of a prompt to Mistral. As is always the case for such a methodology, it is always possible that alternative prompting strategies would yield different performance. In particular, we have provided English language prompts for German text. While Mistral is known to be among the most robust LLMs in handling multiple languages (Holtermann et al., 2024), prompting in the same language as the data under analysis may provide different results. Also, we restricted ourselves to a single run on each experiment.

In addition, the symbolic rules generated by the inductive logic programming process are limited to the templates provided to the system (as shown in figure 5). These rules are currently quite limited, restricted to a small number of key terms across the response, question and grading rubric. However, the initial analysis from spaCy provides extensive additional information on the parsed sentences, such as part of speech, grammatical relations, and so on. Our future work will investigate whether more information in the induced rules can be used to improve overall performance.

Ethics statement

We confirm that this work complies with the [ACL code of ethics](#). No additional data sources were used beyond those cited in this paper, and the data provided by the team for the shared task.

Our interest in this work is driven by our belief that in issues such as grading, which can have an impact on peoples' futures, AI-based decisions must be clear and explainable. Audit trails for grading are a fundamental requirement when AI is used so that results can be challenged. We hope to be part of a movement of explainable AI behaviour in classroom environments.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments on the original draft of this paper, and particularly their suggestions for future directions for our work.

References

Dishank Aggarwal, Pritam Sil, Bhaskaran Raman, and Pushpak Bhattacharyya. 2025. "I Understand Why I Got This Grade": Automatic Short Answer Grading (ASAG) with Feedback. In *Artificial Intelligence in*

Education, pages 304–318, Cham. Springer Nature Switzerland.

Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and T. P. Singh. 2024. [Neuro-symbolic artificial intelligence: a survey](#). *Neural Computing and Applications*, 36(21):12809–12844.

Ian Clark. 2012. [Formative Assessment: Assessment Is for Self-regulated Learning](#). *Educational Psychology Review*, 24(2):205–249.

Andrew Cropper, Sebastijan Dumančić, Richard Evans, and Stephen H. Muggleton. 2022. [Inductive logic programming at 30](#). *Machine Learning*, 111(1):147–172. ArXiv: 2102.10556 ISBN: 0123456789.

Jiaxin Cui and Rohan Alexander. 2026. [Same Prompt, Different Outcomes: Evaluating the Reproducibility of Data Analysis by LLMs](#). ArXiv: 2602.14349.

Marius-Constantin Dinu, Claudiu Leoveanu-Condrei, Markus Holzleitner, Werner Zellinger, and Sepp Hochreiter. 2024. [SymbolicAI: A framework for logic-based approaches combining generative models and solvers](#). *_eprint*: 2402.00854.

Myroslava O. 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.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. 2019. Multi-shot ASP solving with *clingo*. *Theory and Practice of Logic Programming*, 19(1):27–82.

Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berrit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, and Hendrik Drachsler. 2026. Report on the BEA 2026 Shared Task on Rubric-based Short Answer Scoring for German. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.

Carolyn Holtermann, Paul Röttger, Timm Dill, and Anne Lauscher. 2024. Evaluating the Elementary Multilingual Capabilities of Large Language Models with MultiQ. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4476–4494.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*.

- Mark Law, Alessandra Russo, and Krysia Broda. 2014. Inductive Learning of Answer Set Programs. In *European Conference on Logics in Artificial Intelligence (JELIA)*. ArXiv: 1608.01946.
- Joan Lu, Bhavya Krishna Balasubramanian, Mike Joy, and Qiang Xu. 2025. [Survey and Analysis for the Challenges in Computer Science to the Automation of Grading Systems](#). *ACM Computing Surveys*, 58(1).
- Mistral AI. 2024. [Mistral Large 3 Model Card](#).
- Stephen Muggleton. 1995. [Inverse entailment and Prolog](#). *New Generation Computing*, 13(3-4):245–286.
- Stephen Muggleton and Cao Feng. 1990. [Efficient Induction of Logic Programs](#). *New Generation Computing*, pages 368–381. ISBN: 354063875X.
- Suraj Jung Pandey. 2022. [Modelling Alignment and Key Information for Automatic Grading](#). PhD Thesis, The Open University.
- J R Quinlan. 1990. [Learning logical definition from Relations](#). *Machine Learning*, 5(1990):239–266.
- Ayush Thakur. 2025. [Eisvogel: Evaluating German Language Proficiency](#).
- Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. [Solving olympiad geometry without human demonstrations](#). *Nature*, 625(7995):476–482.
- Alistair Willis. 2015. [Using NLP to Support Scalable Assessment of Short Free Text Responses](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 243–253, Denver, Colorado. Association for Computational Linguistics.
- Xiao-Wen Yang, Jie-Jing Shao, Lan-Zhe Guo, Bo-Wen Zhang, Zhi Zhou, Lin-Han Jia, Wang-Zhou Dai, and Yu-Feng Li. 2025. [Neuro-Symbolic Artificial Intelligence: Towards Improving the Reasoning Abilities of Large Language Models](#). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 10770–10778. International Joint Conferences on Artificial Intelligence Organization.

IWM-DKM at BEA 2026 Shared Task 2: Supplementing Supervised Fine-Tuning for Rubric-Based Short Answer Scoring

Kate Rebecca Belcher^{1,2}, Marius De Kuthy Meurers³,
Kordula De Kuthy¹, Detmar Meurers^{1,2}

k.belcher@iwm-tuebingen.de, marius.dkm@tum.de, k.dekuthy@iwm-tuebingen.de, d.meurers@iwm-tuebingen.de

¹Leibniz-Institut für Wissensmedien, ²University of Tübingen, ³Technical University of Munich

Abstract

In this paper, we present the IWM-DKM team submissions to the BEA 2026 Shared Task 2: Rubric-based Short Answer Scoring for German. We systematically explored how fine-tuned language models can be reliably employed for short answer scoring, for which three aspects turn out to be particularly beneficial: supplementing the fine-tuning process with generated domain expertise, restructured rubrics, and thinking traces. To increase the robustness of the scoring, we combine distinct approaches in an ensemble. Our best submissions finished in first place across all tracks, indicating promise for the further application of these strategies in automatic scoring.

relevant information such as *domain-specific background knowledge*, and *restructuring information*, including both the rubrics and the model’s thinking traces to increase grading consistency.

Our approach produced the best submission in both *unseen question* (UQ) tracks with a single fine-tuned model combined with several supplementary strategies, and achieved first place in both *unseen answer* (UA) tracks by combining the strengths of multiple models into a majority-vote ensemble. In the following, we discuss related work on rubric-based short answer scoring and what motivated our use of the techniques we implemented. We then describe our methodology in detail and outline the results for each track.

1 Introduction

In contrast to previous challenges for automatic short answer scoring, such as the 2012 Kaggle challenge (Barbara et al., 2012) and the 2013 SemEval Task 7 (Dzikovska et al., 2013), the BEA 2026 Shared Task 2 (Gombert et al., 2026) represented the first challenge in which *scoring rubrics* formed an essential element of the scoring, and the first large-scale non-English short-answer challenge. The shared task offered two labeling schemes: a 3-way classification of answers into ‘Correct’, ‘Partially correct’ and ‘Incorrect’, and a binary ‘Correct’ / ‘Incorrect’ classification. Additionally, two evaluation setups were provided: *Unseen Answers* (a held-out set of answers to the questions from the training data), and *Unseen Questions* (answers for new questions not included in the training data). Participating as team IWM-DKM, we submitted entries to all four tracks of the challenge.

We utilized LoRA-fine-tuned LLMs (Hu et al., 2022) from the Qwen family (Yang et al., 2025) as the backbone of our grading system and successfully improved performance by a combination of *augmentation* of the model input with additional

2 Related Work

Recent years have seen an increase in deep learning approaches (e.g. Bexte et al., 2022; Ormerod, 2022) and LLM-implementations for short answer scoring (e.g. Chamieh et al., 2024; Frohn et al., 2025; Gao et al., 2025). In rubric-based scoring, descriptive scoring rubrics are integrated into the auto-grading methodology, acting as a mechanism to guide model decisions closer in line with principles of human assessment. Wang et al. (2019) integrate “key elements” of scoring rubrics with a word-level attention mechanism in a Bi-LSTM neural grading system. In recent approaches with LLMs, both Zhao et al. (2024) and Jiang and Bosch (2024) find that LLM scoring reliability is improved with the addition of rubrics, and Frohn et al. (2025) observe variable performance dependent on the level of rubric detail. Somewhat contrastingly, Pathak et al. (2025) experiment with prompt granularity, and find that LLM grading may become too harsh when rubrics are evaluated in a point-wise manner. Pathak et al. develop a multi-agent LLM approach, utilizing a majority vote ensemble of models. While they observe benefits of an ensem-

ble over a single model, beyond 3 to 4 LLMs results plateau with the addition of more LLMs. Overall, works to date suggest that rubrics offer a promising way to improve performance in short answer scoring, with the additional benefit of greater grounding in principles of traditional human assessment. However, how best to incorporate rubrics into automatic grading has not yet been comprehensively explored. While many recent approaches integrate rubric information via LLM prompting, the extent to which rubrics can be applied to the fine-tuning of pre-trained LLMs remains unclear. With our shared task submission, we aim to target this gap.

3 Approach

The ALICE-LP dataset consists of questions and answers from four science subject areas with authentic responses from secondary school students in Germany. The training set contained 78 questions with 7899 answers, alongside a trial set of 827 answers. Each item contains the question, the student response, and a rubric describing the criteria for each scoring level. Preliminary experiments with several encoder models appeared to reach a performance ceiling at around 0.75 QWK on the 3-way trial set. Consequently, our primary approach instead utilized supervised fine-tuning of decoder-only language models, specifically Qwen3 (4B/8B) and Qwen3.5 (9B) (Yang et al., 2025), incorporating all three item elements into the model input. We opted to use Qwen models, as relatively small-sized yet high performing open-source LLMs, making them more suitable for fine-tuning and deployment in educational settings than closed-source alternatives (Lin et al., 2026).

We employed Low-Rank Adaptation (LoRA) (Hu et al., 2022) targeting all linear projection layers with a rank $r = 16$ and a scaling factor $\alpha = 16$. This fine-tuning strategy was chosen so as to preserve the underlying model as much as possible whilst also taking into account the nature of the task-dataset. Models were trained using the AdamW optimizer (Loshchilov and Hutter, 2017) with a peak learning rate between 1×10^{-4} and 2×10^{-4} and a global batch size of 16 or 32. We applied a cosine learning rate schedule to smoothly decay the learning rate and trained the models for up to 10 epochs, ultimately selecting the best checkpoint based on the highest validation Quadratic Weighted Kappa (QWK). For certain models, we experiment with different fine-tuning

strategies, including Direct Preference Optimization (DPO) (Rafailov et al., 2023) (where the probabilities of correct labels are increased relative to dis-preferred options); Fast Gradient Method (FGM) (Goodfellow et al., 2014; Miyato et al., 2017) (an adversarial training method whereby embeddings are perturbed to increase robustness); and NEFTune (Jain et al., 2023) (a further method to introduce noise to the vector embeddings). We indicate in Section 4 for which submissions we employed alternative fine-tuning strategies. We build on this foundation with a range of approaches that aim to specialize the model more closely to the domain of the task and align the model’s decision process with the information contained in the rubrics. We do this by providing the model with *more* relevant information and by restructuring the format of it.

3.1 Checklist Thinking

Based on the principle of walking through intermediary steps before giving a final answer, the idea of “thinking” in LLMs has evolved from enhancing prompts with chain-of-thought prompting (Kojima et al., 2022) to integrating thinking steps into the model fine-tuning process (Zelikman et al., 2022), with the aim of improving model consistency. To apply this principle to short answer grading, we introduce a structured reasoning format termed *Checklist Thinking*. Instead of predicting the score label directly, the model is trained to first output a concise set of “Yes/No” checks corresponding to each rubric level within `<think>` tags. We found that a dynamic “Applicable-last” order was most effective—where the model explicitly evaluates and rejects the non-applicable score categories before confirming the criteria for its final predicted label. This method was particularly advantageous in improving recall for the “Partially correct” category, an example of which is:

```
<think>
Incorrect: NEIN
Correct: NEIN
Partially correct: JA
</think>
Partially correct
```

3.2 Rubric Reframing

LLMs typically perform better on tasks that use a structured, closed format. In question answering tasks, a performance drop is observed when moving from multiple choice to open-question tasks (e.g. Li et al., 2024). Several works address this potential challenge for LLMs by breaking down

complex tasks into a series of sub-tasks (Dua et al., 2022; Qin et al., 2024; Zhou et al., 2023), or by decomposing instructions into Yes/No checklists, resulting in greater alignment with human preference (Cook et al., 2024). Taking inspiration from these techniques, we consider how the format of the rubrics can be restructured to be more accessible to LLMs. For each 3-way rubric, we prompt GPT5.1 (Singh et al., 2025) to transform the rubric from high-level descriptions of the categorical levels to a set of questions that can be followed as a decision tree, providing greater structure to guide the LLM to a grading decision. The prompt used to generate the reformulated rubrics is shown in Appendix C.1 and an example is shown in Appendix B.

3.3 Background Knowledge

A characteristic of the dataset is that many questions require an understanding of subject-specific terminology and processes, or relate to external material (e.g., tables, experiments). To provide the necessary domain expertise, and counter some of the lack of context present in the question alone, we supplement the model input with generated “background knowledge”. We use Claude Sonnet 4.6 (Anthropic, 2026b) to generate a short textbook-style summary based on the question, the rubric, and all examples of correct student answers available for the given question. The exact number of examples used to generate background knowledge was variable, with an average of 26.46 correct answers per question in the unseen answers train set. For *unseen questions*, where ground-truth labels are absent, we employ a two-step pipeline: We initially generate context from all answers, from which labels are then predicted, and then re-generate background knowledge in a second pass using only the answers labelled as “Correct” (i.e., ‘silver-standard’ labels) (see Appendix C.2 for prompt details, and Appendix B for an example).

3.4 Skilled-based Categories

To better understand the nature of the data, and due to the breadth of question types in the dataset, we manually annotated each question with the operator used in the question (e.g., *describe*, *justify*, *explain* etc.). In addition, we assign each question to one of four categories according to the skills targeted in the question: *Concept Description* (describing abstract concepts or processes), *Example Led* (analyzing examples in relation to fundamental concepts), *Idea Seeking* (the formulation of hypotheses

or examples), and *Mathematical* (calculations and graph interpretations). The conceptual basis for the categories was established in discussion among the authors and annotation was subsequently carried out by one of the authors. An example question for each category can be found in Appendix D. We use these categorizations in two ways. Firstly, motivated by the idea that scoring performance may be variable in accordance with question type (e.g. Meurers et al., 2011), we investigate whether fine-tuning individual models based on questions of a similar nature (i.e., of the same skill category) improves performance for UA. At inference, predictions are made using the corresponding skill-based model. Secondly, we use these categories as the basis to create a custom development set for UQ, including questions which are representative of the skills and operators present in the UQ test set. We experiment between a manually chosen subset of 10 questions (n=1488), and a broader subset of 20 questions (n=1368), selected interactively with Claude Opus 4.6 (Anthropic, 2026a).

3.5 Ensemble Approach

Ensembles of multiple models have been successfully used for short answer scoring in several instances (Ormerod, 2022; Pathak et al., 2025). Majority voting of a number of individual models can help to mitigate limitations of single models, leading to more robust overall scoring. We utilize this majority vote approach in the UA tracks. To select an optimal combination, we evaluated a candidate pool of 100 checkpoints from our top experimental runs, based on QWK on the 3-way trial set. We performed a brute-force search over these candidates to identify the best combination of models for the ensemble. In line with Pathak et al. (2025), we use a combination of four models in the ensemble.

4 Results

We selected our test set submissions based on validation performance, using the provided trial set for unseen answers and custom train/validation splits for unseen questions. We aimed for a balanced representation of the strategies investigated during development. Official test results are in Table 1.

4.1 Unseen Answers

For the UA tracks, we tested fine-tuned Qwen3 (4B/8B) and Qwen3.5-9B models. During development, our best performing single model on the 3-way *trial* set was Qwen3-4B (0.792 QWK). Thus,

Rank	Submission ID	Base Model	Additional Details	QWK	Precision	Recall	F1
3-way Unseen Answers							
1	655003	Multi-Model	Ensemble: 1x9B, 2x8B, 1x4B	0.796	0.781	0.782	0.78
2	655652	Qwen3-8B	+FGM, +CL, +BG_correct, +full_data	0.79	0.772	0.77	0.771
8	648459	Qwen3-4B	<i>no special parameters</i>	0.781	0.779	0.767	0.77
9	654909	Qwen3-4B	by_skill	0.78	0.775	0.768	0.77
10	654389	Qwen3-8B	+CL, +BG_correct	0.779	0.771	0.763	0.765
2-way Unseen Answers							
1	655003	Multi-Model	Ensemble: 3→2way	0.726	0.887	0.887	0.887
4	654846	Multi-Model	Ensemble: Rank 7, 8, 10 submissions	0.71	0.88	0.881	0.881
7	654838	Qwen3-4B	by_skill, 3→2way	0.7	0.876	0.877	0.876
8	653676	Qwen3-4B	3→2way	0.698	0.875	0.877	0.876
10	653641	Qwen3-4B	2way	0.684	0.869	0.87	0.869
3-way Unseen Questions							
1	655694	Qwen3.5-9B	+NEFT, +CL, +BG_correct, +RR, custom_UQ_20	0.681	0.68	0.664	0.669
6	654994	Qwen3.5-9B	+NEFT, +CL, +BG_all, +RR, custom_UQ_20	0.635	0.657	0.64	0.644
13	653605	Qwen3.5-9B	+BG, +RR, +all_data, custom_UQ_10	0.591	0.727	0.601	0.6
16	654861	Qwen3.5-9B	+BG, +RR, custom_UQ_10	0.541	0.738	0.571	0.567
2-way Unseen Questions							
1	655727	Qwen3.5-9B	+NEFT, +CL, +BG_correct, +RR, custom_UQ_20, 3→2way	0.55	0.813	0.818	0.815
5	655638	Qwen3-4B	UA train, 2way	0.526	0.802	0.797	0.799
10	655654	Qwen3.5-9B	+NEFT, +CL, +BG_all, +RR, custom_UQ_20, 3→2way	0.501	0.796	0.804	0.797

Table 1: Test set results for IWM-DKM submissions for all four tracks. Abbreviations: BG = background knowledge (data source), CL = checklist thinking, RR= reframed rubrics, NEFT = NEFTune, by_skill = skill-specific classifiers, custom_UQ = custom validation set (with total questions), 3→2way = mapping 3-way label predictions to 2-way labels, UA train = trained only on Unseen Answers data, FGM = Fast Gradient Method adversarial training, full data = validation set included in training. Precision, Recall and F1 are all weighted metrics.

we submitted the 4B model, trained on either 3-way or 2-way labels, as baselines for the corresponding labeling scheme. We tested further strategies as described in Section 3, individually and in ensembles.

4.1.1 3-way Unseen Answers

The Qwen3-4B model achieved 8th place with a QWK on the test set of 0.781, providing a strong baseline. Our variations to this, such as fine-tuning individual Qwen3-4B models for each skill-based category (Rank 9), and including generated background knowledge and checklist thinking with the 8B model (Rank 10), performed on par, but were unable to beat the baseline. However, a variation of the Rank 10 submission that included trial data in training and was fine-tuned with the Fast Gradient Method (Miyato et al., 2017) demonstrated the strength of this form of adversarial training, placing second on the leaderboard. Our best-performing and final submission (0.796 QWK) was a majority-vote ensemble of four Qwen-based variations. Details of the models in this ensemble are shown in Appendix A. The ensemble was particularly effective at resolving ambiguous boundary cases between “Partially correct” and “Correct”.

4.1.2 2-way Unseen Answers

We submitted the two variations of the baseline model (Qwen3-4B) to the 2-way track: firstly, mapping of the 3-way prediction labels to the 2-way la-

bels (i.e., “Partially correct” is re-mapped to “Incorrect”); and secondly, fine-tuning directly on 2-way labels. Whilst both were strong baselines, the 2-way fine-tuned model performed marginally worse than the mapped version (0.684 QWK compared to 0.698 QWK). This suggested an advantage of making predictions at a more fine-grained level, thus, our remaining submissions for the 2-way UA track were mapped from 3-way labels. Mapping predictions from our by-skill Qwen3-4B classifiers led to marginal improvement over the baselines, and a majority-vote ensemble of the three systems ranked 7, 8 and 10 offered another slight performance improvement. However, the more architecturally diverse ensemble (mapped predictions from Rank 1 on the 3-way UA) resulted in our strongest submission, and the strongest overall submission for this track, achieving a QWK of 0.726, once again highlighting the robustness of ensemble setups.

4.2 Unseen Questions

4.2.1 3-way Unseen Questions

For the 3-way UQ track, we utilized Qwen3.5-9B as our base model, supplemented by checklist thinking, reframed rubrics, and background knowledge, and trained with a custom validation set. Submissions in Rank 1 and 6 were additionally fine-tuned with NEFTune (Jain et al., 2023). Between these submissions, we observe that silver

standard “Correct-only” student answers (Rank 1) provided a +0.046 QWK improvement over using all answers for Background Knowledge generation (Rank 6), underscoring that high-quality silver context is markedly more effective in driving generalization than a larger, noisier data pool, where misconceptions may be introduced. Furthermore, we experimented with two custom validation sets and saw a significant improvement in performance with the broader validation set (custom_UQ_20, used in Rank 1 and 6) compared to the 10 question set (custom_UQ_10, used for Rank 13 and 16¹). Despite strong performance during development, the model optimized on the narrower, 10-question validation set showed poor transfer abilities to the test set, indicating the advantage of a broader pool of validation answers to aid reliable generalization.

4.2.2 2-way Unseen Questions

We applied several of our submissions from other tracks to the 2-way UQ track. To test generalizability from *unseen answers* to *unseen questions*, we evaluate Qwen3-4B fine-tuned on 2-way labels for UA on the 2-way UQ test set. Despite only training and validating on unseen answers, the model placed fifth on the leaderboard, although with a reduction in performance of 0.158 QWK (0.07 weighted F1). Additionally, we map the labels of our top two submissions from 3-way UQ, and reinforcing our earlier findings on the importance of high-quality background knowledge, the model trained with NEFTune, reframed rubrics, checklist thinking, the custom_UQ_20 validation set, and background knowledge only outperformed the baseline when background knowledge was generated from silver-labels. On the first pass, where background knowledge was generated from all available student answers, overall performance was worse than our baseline of Qwen3-4B.

5 Discussion

Our results show that the outlined supplementary strategies improve fine-tuning performance. The results for both UQ tracks indicated the importance of *high-quality* additional information. Our leading submissions contained background knowledge generated from “Correct” student answers (as predicted by a first pass model), whereas results for an otherwise identical model with background knowledge based on all students answers performed sig-

nificantly worse. We also observed during development that individual techniques taken alone can actually be *detrimental* to performance, their positive contribution only arises when combined.

We observed mixed results when it came to the optimization of the validation set for UQ. A broader 20-question trial set offered substantial improvements over a narrower 10-question trial set. Somewhat conflictingly, the Qwen3-4B model only optimized for unseen answers also exhibited promising performance in the 2-way UQ track, leading to further questions about the most impactful strategies to improve generalization in short answer grading.

Broadly, simple fine-tuning of decoder-only models was highly effective. Encoder-based models have generally dominated the short-answer scoring literature to date (e.g., Bexte et al., 2022; Sung et al., 2019), but despite experimenting with fine-tuning several German- and multilingual BERT-based models (e.g., ModernGBERT (Wunderle et al., 2025), gelectra-large (Chan et al., 2020)) in earlier trials, decoder-only architectures resulted in stronger performance. This reflects the NLP shift towards decoder-only architectures (Zhang et al., 2025; BehnamGhader et al., 2024), whose richer contextual embeddings better handle the nuanced semantics of rubric-based scoring. Finally, model ensembling proved beneficial, particularly in resolving boundary ambiguities between adjacent scoring categories. This is particularly advantageous in light of the fact that fine-tuned models exhibit sensitivity to small differences such as individual epochs and differences in seed. However, the high computational cost of this strategy limits the suitability of ensembles in real-life scoring.

6 Conclusion

Overall, our shared task submissions demonstrated the potential for fine-tuned decoder-only LLMs for rubric-based short answer scoring. Not only was LoRA fine-tuning of relatively small LLMs effective, but we highlighted how performance can be further improved by supplementing the input with task-relevant information. Our approach provides the model with more relevant information and presents that information in multiple structured formats to obtain more robust results. In future work, it would be interesting to further investigate the contributions of the individual strategies we used and to systematically evaluate their impact in other scoring contexts.

¹A formatting mismatch in the reframed rubrics used in the Rank 16 submission may have also influenced performance.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under Germany’s Excellence Strategy – EXC number 2064/1 – Project number 390727645. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the DFG through grant no. INST 37/1057-1 FUGG (Tübingen Machine Learning Cloud) and thank the Joachim Herz Stiftung for its support through the ALEE project.

Limitations

Due to the wide variety of methods we tested in the development phase, we were not able to systematically assess the impact of each individual technique on test set performance, nor did we evaluate all the models in our leading ensemble approaches as individual entries. Similarly, we did not test all our approaches on all tracks, which limits our ability to draw conclusions regarding the generalizability of methods across testing set-ups. Nevertheless, we believe our findings suggest support for the use of supervised fine-tuning of decoder-only models with LoRA in educational contexts. We used a single annotator for the skill-based categories. Double-annotation would strengthen the reliability of the categorization approach. In future work, it may be beneficial to test the techniques we implemented to improve upon standard fine-tuning on other datasets, and in more limited data scenarios, and to evaluate how expert knowledge could be used to further improve those techniques which were reliant on synthetic data (e.g., generated background knowledge, synthetically reframed rubrics).

Ethical Considerations

The use of AI for educational purposes has been designated as ‘high risk’ by the EU AI Act. All the data used in this shared task are anonymous and contain no personal identifying information, however, this remains an important consideration when thinking about the deployment of AI-based tools in real-world educational contexts (e.g., in classrooms). We acknowledge the use of AI tools for the generation of code in developing our systems, and for some of our data augmentation strategies, which carries both monetary and environmental costs.

References

- Anthropic. 2026a. [System card: Claude Opus 4.6](#). Technical report.
- Anthropic. 2026b. [System card: Claude Sonnet 4.6](#). Technical report.
- Barbara, Ben Hamner, Jaison Morgan, lynnvande, and Mark Shermis. 2012. The hewlett foundation: Short answer scoring. <https://kaggle.com/competitions/asap-sas>. Kaggle.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [Llm2vec: Large language models are secretly powerful text encoders](#). In *First Conference on Language Modeling, COLM 2024*.
- 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.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. [LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 309–315, Mexico City, Mexico. Association for Computational Linguistics.
- Branden Chan, Stefan Schweter, and Timo Möller. 2020. German’s next language model. In *Proceedings of the 28th international conference on computational linguistics*, pages 6788–6796.
- Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. 2024. [Ticking all the boxes: Generated checklists improve llm evaluation and generation](#). *Preprint*, arXiv:2410.03608.
- Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. [Successive prompting for decomposing complex questions](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1251–1265, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- 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.

- Scott Frohn, Tyler Burleigh, and Jing Chen. 2025. Automated scoring of short answer questions with large language models: Impacts of model, item, and rubric design. In *Artificial Intelligence in Education*, pages 44–51, Cham. Springer Nature Switzerland.
- Xunyi Gao, Shamyia Karumbaiah, Adi Dalal, Indrani Dey, Dana Gnesdilow, and Sadhana Puntambekar. 2025. A comparative analysis of llm and specialized nlp system for automated assessment of science content. In *Artificial Intelligence in Education*, pages 76–82, Cham. Springer Nature Switzerland.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, and Hendrik Drachsler. 2026. Report on the bea 2026 shared task on rubric-based short answer scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.
- Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, and 1 others. 2023. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*.
- Lan Jiang and Nigel Bosch. 2024. [Short answer scoring with GPT-4](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, pages 438–442, Atlanta GA USA. ACM.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Wangyue Li, Liangzhi Li, Tong Xiang, Xiao Liu, Wei Deng, and Noa Garcia. 2024. [Can multiple-choice questions really be useful in detecting the abilities of LLMs?](#) In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2819–2834, Torino, Italia. ELRA and ICCL.
- Michael Pin-Chuan Lin, Jing-Yuan Huang, Daniel H. Chang, Gerald Tembrevilla, G. Michael Bowen, Eric Poitras, Vasudevan Janarthanan, and Jeeho Ryoo. 2026. [Open-source large language models in education: A narrative review of evidence, pedagogical roles, and learning outcomes](#). *AI in Education*, 2(1).
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp. 2011. [Evaluating answers to reading comprehension questions in context: Results for German and the role of information structure](#). In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment*, pages 1–9, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. [Adversarial training methods for semi-supervised text classification](#). In *International Conference on Learning Representations (ICLR)*.
- Christopher Ormerod. 2022. [Short-answer scoring with ensembles of pretrained language models](#). *arXiv preprint*. Version Number: 1.
- Aditya Pathak, Rachit Gandhi, Vaibhav Uttam, Arnab Ramamoorthy, Pratyush Ghosh, Aaryan Raj Jindal, Shreyash Verma, Aditya Mittal, Aashna Ased, Chirag Khatri, Yashwanth Nakka, Devansh, Jagat Sesh Challa, and Dhruv Kumar. 2025. [Rubric is all you need: Improving llm-based code evaluation with question-specific rubrics](#). In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1, ICER '25*, page 181–195, New York, NY, USA. Association for Computing Machinery.
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. [InFoBench: Evaluating instruction following ability in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.
- 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. *Advances in neural information processing systems*, 36:53728–53741.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267. Version 5.1.
- Chul Sung, Tejas I. Dhamecha, and Nirmal K. Mukhi. 2019. [Improving Short Answer Grading Using Transformer-Based Pre-training](#). In *International Conference on Artificial Intelligence in Education*.
- Tianqi Wang, Naoya Inoue, Hiroki Ouchi, Tomoya Mizumoto, and Kentaro Inui. 2019. [Inject rubrics into short answer grading system](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for*

Low-Resource NLP (DeepLo 2019), pages 175–182, Hong Kong, China. Association for Computational Linguistics.

Julia Wunderle, Anton Ehrmanntraut, Jan Pfister, Fotis Jannidis, and Andreas Hotho. 2025. New encoders for german trained from scratch: Comparing modernngbert with converted llm2vec models. *arXiv preprint arXiv:2505.13136*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: self-taught reasoner bootstrapping reasoning with reasoning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.

Biao Zhang, Yong Cheng, Siamak Shakeri, Xinyi Wang, Min Ma, and Orhan Firat. 2025. [Encoder-decoder or decoder-only? revisiting encoder-decoder large language model](#). *Preprint*, arXiv:2510.26622.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. [Retrieval-Augmented Generation for AI-Generated Content: A Survey](#). *ArXiv*, abs/2402.19473.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

A Ensemble model details

We list the details of the models included in our four-model ensemble for 3-way unseen answers submission in Table 2.

Base model	Details
Qwen3.5-9B	+BG_correct +CL
Qwen3-8B	+FGM, +BG_correct, +CL
Qwen3-8B	+DPO
Qwen3-4B	+BG_correct, +CL

Table 2: Models included in ensemble for Rank 1 Unseen Answers 3-way submission (Abbreviations: BG_correct = Background Knowledge from correct answers, CL = checklist thinking, FGM = Fast Gradient Method, DPO = Direct Preference Optimization)

B Example Item with Background Knowledge and Reframed Rubric

Table 3 shows an example item from the ALICE-LP dataset, illustrating the question, generated background knowledge, original scoring rubric, reframed rubric, operator, and skill category. The background knowledge field contains the generated passage used as supplementary material during fine-tuning (Section 3.3). This item is representative of the *Concept Description* skill category, with the operator *Beschreiben* (‘Describe’) (Section 3.4).

C Prompts

C.1 Prompt for Reframed Rubrics

The prompt used with GPT5.1 to generate reformulations of the 3-way rubrics in decision tree format is shown in Listing 1.

C.2 Prompt for Background Knowledge

The prompt used with Claude Sonnet for generating question-specific background knowledge is provided in Listing 2.

D Skill-based categories

Table 4 shows an example question for each of the four categories we annotate the data with, as described in (Section 3.4).

Category	Example
Question	Beschreibe das Power-to-Gas Verfahren in eigenen Worten. 'Describe the power-to-gas process in your own words.'
Background Knowledge	Beim Power-to-Gas-Verfahren wird überschüssige elektrische Energie, die zum Beispiel durch Windkraftanlagen erzeugt wird, in chemische Energie umgewandelt und so langfristig gespeichert. Dazu durchläuft die Energie eine Kette von Umwandlungsschritten: Zunächst wird Bewegungsenergie des Windes im Windrad in elektrische Energie umgewandelt, die dann einem Elektrolyseur zugeführt wird. Im Elektrolyseur wird mithilfe der elektrischen Energie Wasser in seine Bestandteile Wasserstoff und Sauerstoff aufgespalten, wobei die Energie in Form von chemischer Energie im Wasserstoff gespeichert wird. Der erzeugte Wasserstoff kann anschließend in Gasspeichern oder im Erdgasnetz gelagert werden, bis er bei Bedarf in einem Gaskraftwerk wieder in elektrische Energie umgewandelt wird. Die vollständige Energieumwandlungskette lautet also: Bewegungsenergie → elektrische Energie → chemische Energie (Wasserstoff) → elektrische Energie. 'In the power-to-gas process, excess electrical energy—generated, for example, by wind turbines—is converted into chemical energy and thus stored for the long term. To achieve this, the energy undergoes a series of conversion steps: First, the kinetic energy of the wind is converted into electrical energy in the wind turbine, which is then fed into an electrolyzer. In the electrolyzer, electrical energy is used to split water into its components, hydrogen and oxygen, with the energy being stored in the hydrogen in the form of chemical energy. The hydrogen produced can then be stored in gas storage facilities or in the natural gas grid until it is converted back into electrical energy in a gas-fired power plant when needed. The complete energy conversion chain is therefore: kinetic energy → electrical energy → chemical energy (hydrogen) → electrical energy.'
Rubric	(Incorrect) Die SuS formulieren keine korrekte Energieumwandlungskette. 'The students do not formulate a correct energy conversion chain.' (Partially correct) Die SuS formulieren eine teils korrekte Energieumwandlungskette (mindestens die Hälfte) 'The students formulate a partially correct energy conversion chain (at least half of it)' (Correct) Die SuS formulieren die vollständige Energieumwandlungskette. 'The students describe the complete energy conversion chain.'
Reframed Rubric	1. Beschreibt die Antwort überhaupt eine Energie-Umwandlungskette beim Power-to-Gas-Verfahren? 'Does the answer even describe an energy conversion chain in the power-to-gas process?' - **Nein** → **Incorrect**, **Ja** → 2 2. Wird deutlich, dass elektrische Energie (z.B. aus erneuerbaren Quellen) zunächst in chemische Energie von Wasserstoff umgewandelt wird (Elektrolyse von Wasser)? 'Is it clear that electrical energy (e.g., from renewable sources) is first converted into the chemical energy of hydrogen (through the electrolysis of water)?' - **Nein** → **Incorrect**, **Ja** → 3 3. Wird zusätzlich beschrieben, dass aus dem Wasserstoff ein gasförmiger Energieträger (z.B. Methan) entsteht, der gespeichert / ins Gasnetz eingespeist werden kann? 'Is it also described that hydrogen is converted into a gaseous energy carrier (e.g., methane) that can be stored or fed into the gas grid?' - **Nein** → - Wenn nur ein Teil der Umwandlungskette korrekt ist (z.B. Strom → Wasserstoff, aber ohne weitere Verwendung / Methanisierung) 'If only part of the conversion chain is correct (e.g., electricity → hydrogen, but without further use or methanation)' → **Partially correct** - **Ja** → 4 4. Ist die gesamte Energieumwandlungskette inhaltlich richtig und in der Reihenfolge vollständig beschrieben (z.B. erneuerbare elektrische Energie → Elektrolyse → Wasserstoff → ggf. Methanisierung → Speicherung / Nutzung im Gasnetz)? 'Is the entire energy conversion chain described accurately contentwise and completely in terms of sequence (e.g., renewable electrical energy → electrolysis → hydrogen → methanation, if applicable → storage/use in the gas grid)?' - **Nein** → **Partially correct**, **Ja** → **Correct**
Operator	Beschreiben ('describe')
Skill Category	Concept Description
Example Student Answer	Die Elektrische Energie wird in Chemische umgewandelt welche dann wieder zu Elektrischer wird. 'Electrical energy is converted into chemical energy, which is then converted back into electrical.'
Gold Label	Partially correct

Table 3: Example item from the ALICE-LP dataset. The example student answer correctly identifies two of the three conversion steps (electrical → chemical → electrical) but omits the initial step (kinetic energy → electrical energy via wind turbine), warranting a *Partially correct* label. (Original is in German; English translation in quotes added here for transparency.)

Sie sind pädagogische/r Assistent/in und haben die Aufgabe, leicht verständliche Bewertungsrubriken für Fragen in formativen Schülerbeurteilungen zu entwerfen. \ Der Bildungskontext sind deutsche Gemeinschaftsschulen und Gymnasien und die abgedeckten Fachbereiche sind: Biologie, Chemie, Physik und Mathematik.

'You are a pedagogical assistant, whose goal is to help design easy-to-follow grading rubrics to questions in formative student assessments. \ The educational setting is German Gemeinschaftsschulen and Gymnasien and the subject domains covered are: Biology, Chemistry, Physics and Mathematics.'

Ihnen wird eine Frage und eine bestehende Rubrik mit drei Kategorien 'Incorrect', 'Partially correct' und 'Correct' vorgelegt. \ Ihre Aufgabe besteht darin, die Rubrik im Format eines Entscheidungsbaums mit Fragen neu zu schreiben, um die richtige Note zuverlässiger zu vergeben. Stellen Sie sicher, dass alle relevanten Informationen aus der bestehenden Rubrik in den Entscheidungsbaum aufgenommen werden.

'You will be presented with a question, and an existing rubric containing three categorical levels 'Incorrect', 'Partially correct' and 'Correct'. \ Your task is to re-write the rubric in the format of a decision tree of questions to assign the correct grade more reliably. Make sure all the relevant information from the existing rubric is included in the decision tree.'

Hier ist die Frage: {question}
 'Here is the question: {question}'

 Hier ist die bestehende Rubrik.
 'Here is the existing rubric.'
 {rubric}

 Neu gestaltete Entscheidungsbaum-Benotungshilfe:
 'Newly designed decision tree grading aid:'

Listing 1: Prompt used to generate reframed rubrics with GPT5.1 (Original is in German; English translation in quotes added here for transparency.)

Schreibe einen Lernzettel-Eintrag (3-5 Sätze Fliesstext, kein Markdown) zu dem naturwissenschaftlichen Thema, das in der folgenden Aufgabe behandelt wird. Der Lernzettel soll die wichtigsten Fachbegriffe, Konzepte und Zusammenhänge erklären, die man kennen muss, um die Aufgabe vollständig beantworten zu können. Schreibe so, als würdest du einem Mitschüler das nötige Vorwissen kompakt zusammenfassen.

'Write a study note entry (3-5 sentences of continuous text, no Markdown) on the scientific topic covered in the following task. The study note should explain the most important technical terms, concepts, and relationships that one needs to know in order to fully answer the task. Write as if you were compactly summarizing the necessary prior knowledge for a classmate.'

Aufgabe: {question}
 'Task: {question}'

Zur Orientierung, welche Konzepte relevant sind:
 'For orientation on which concepts are relevant:'
 - Incorrect: {rubric_incorrect}
 - Partially correct: {rubric_partial}
 - Correct: {rubric_correct}

Korrekte Schülerantworten ({n} Beispiele):
 'Correct student answers ({n} examples):'
 - {Student answer 1}
 - {Student answer 2}
 {...}
 - {Student answer n}

Listing 2: Prompt used to generate background knowledge with Claude Sonnet (Original is in German; English translation in quotes added here for transparency.)

Category	Example
Concept Description	Beschreibe auf Grundlage Deiner oben getroffenen Auswahl in eigenen Worten, was Du allgemein unter der Reaktionsgeschwindigkeit chemischer Reaktionen verstehst. <i>'Based on the selection you made above, describe in your own words what you generally understand by the reaction rate of chemical reactions'</i>
Example Led	Beurteile das Zerteilen der Äpfel hinsichtlich der Geschwindigkeit des Verderblichkeitsprozesses (Hinweis: wenn Du fachliche Informationen zu chemischen Abläufen des Verderbens einbeziehen möchtest, kannst Du diese unter Angabe der Quellen recherchieren und einbinden). <i>'Evaluate how cutting apples affects the rate at which they spoil (Note: If you wish to include technical information on the chemical processes involved in spoilage, you may research and incorporate this information, provided you cite your sources).'</i>
Idea Seeking	Hast Du auf Grundlage der oben beantworteten Aufgaben bereits Ideen, welche Größen geeignet sein könnten, um die Geschwindigkeit dieser Reaktion zu erfassen? Wenn ja, beschreibe und begründe Deine Ideen. <i>'Based on your answers to the questions above, do you already have any ideas about which quantities might be suitable for measuring the rate of this reaction? If so, describe and explain your ideas.'</i>
Mathematical	In 2.2 Untersuchung der Lampen 4 habt ihr die Formel zur Berechnung der elektrischen Leistung genannt. Berechnet mit Hilfe der Formel die elektrischen Leistungen der einzelnen Lampen. <i>'In Section 2.2, "Examination of the Lamps 4," you were given the formula for calculating electrical power. Use this formula to calculate the electrical power of each lamp.'</i>

Table 4: Examples of questions for each skill-based category (Original in German; English translation in quotes added here for transparency)

RETUYT-INCO at BEA 2026 Shared Task 2: Meta-prompting in Rubric-based Scoring for German

Ignacio Sastre[†] and Ignacio Remersaro

Facundo Díaz and Nicolás de Horta

Luis Chiruzzo and Aiala Rosá and Santiago Góngora

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

Abstract

In this paper, we present the RETUYT-INCO participation at the BEA 2026 shared task “Rubric-based Short Answer Scoring for German”. Our team participated in track 1 (*Unseen answers three-way*), track 3 (*Unseen answers two-way*) and track 4 (*Unseen questions two-way*). Since these tracks required scoring short student answers using specific rubrics, we looked for ways to handle the changing nature of the task. We created a method called **Meta-prompting**. In this approach, an LLM creates a custom prompt based on examples from the Train set. This prompt is then used to grade new student answers. Along with this method, we also describe other approaches we used, such as classic machine learning, fine-tuning open-source LLMs, and different prompting techniques. According to the official results, our team placed 6th out of 8 participants in Track 1 with a QWK of 0.729. In Track 3, we secured 4th place out of 9 with a QWK of 0.674, and we also placed 4th out of 8 in Track 4 with a QWK of 0.49.

1 Introduction

Short-answer scoring is a promising research direction in the field of Natural Language Processing for education. Unlike multiple-choice tests, which only require students to recognize a correct answer, short answers require them to actively construct a response using the concepts they have learned from the curriculum. This may be appropriate for capturing a student’s true understanding of a subject. The automation of short-text assessment has the potential to deliver faster feedback and consistent grading, offering vital support to overburdened educators, especially in under-resourced environments. Over the years, researchers have used various benchmarks and shared tasks to refine these

systems (Burrows et al., 2015; Bai and Stede, 2023; Dzikovska et al., 2013). We, as a research group, are deeply interested in these applications of NLP tools (Chiruzzo et al., 2022; Rosá et al., 2025).

In this paper, we present the RETUYT-INCO participation in the BEA 2026 Shared Task 2: Rubric-based Short Answer Scoring for German (Gombert et al., 2026). This task involves the automatic evaluation of short-answer responses in the domains of science and mathematics, framed as a classification problem with both two-way and three-way scoring rubrics. Our team participated in Tracks 1 (Unseen answers three-way), 3 (Unseen answers two-way) and 4 (Unseen questions two-way). Our team has participated in various shared tasks of the BEA workshop in recent years (Baladón et al., 2023; Sastre et al., 2024; Góngora et al., 2025), generally exploring solutions that do not require extensive computational resources. Additionally, this year we were able to incorporate state-of-the-art Gemini models into our experiments through the *Gemini Academic Program*.

2 Meta-prompting

We mainly focus on the *unseen answer* tracks, in which systems must score new student answers for question-rubric pairs that have already been observed in the training data. An analysis of the Train split revealed 72 distinct question-rubric groups, each defined by a fixed question and its associated rubric. As a baseline, we used Gemini 3 Flash¹ with a generic scoring prompt instantiated with the corresponding question and rubric. On the Trial set, this baseline exhibited substantial performance variation across groups, suggesting that a single generic prompt does not capture the scoring criteria equally well for all question-rubric pairs.

¹<https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/>

[†] Corresponding author: isastre@fing.edu.uy.

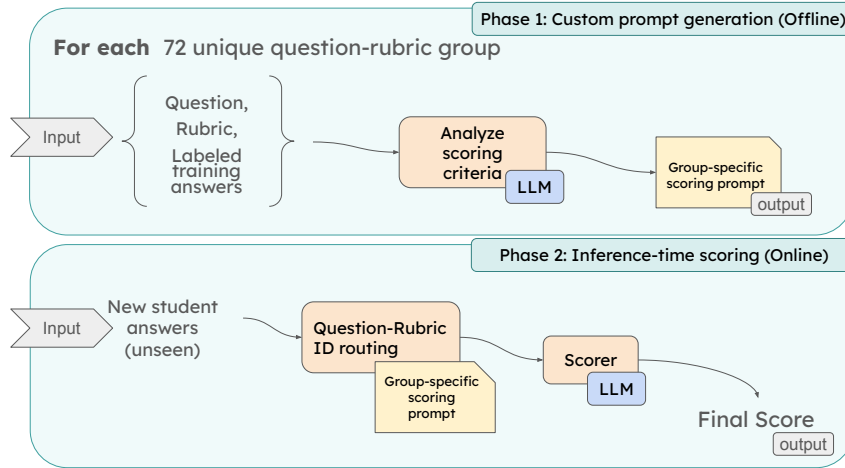


Figure 1: Meta-prompting operation scheme in its two phases

To address the performance inconsistencies observed with generic scoring prompts, we leverage established advances in automated prompt optimization. This field has progressed from Instruction Induction (Honovich et al., 2022), which identifies task descriptions from few-shot examples, to the Automatic Prompt Engineer (Zhou et al., 2023) framework for systematic search and scoring. Most recently, GEPA (Agrawal et al., 2026) introduces a reflexive evolution approach that uses genetic algorithms and natural language reasoning to iteratively refine instructions, offering a robust alternative to static baselines for the diverse requirements of rubric-based scoring.

Based on this line of work, we introduce the **meta-prompting** approach, that induces a custom scoring prompt for each question-rubric group. The method consists of two phases. In the first phase, for each group, we prompt an LLM with the question, the rubric, and the full set of labeled training answers for that group, serialized as JSON. The model is asked to generate a reusable scoring prompt tailored to that specific group. In the second phase, at inference time, each new student answer is routed to its corresponding question-rubric group and evaluated using the group-specific prompt generated in the first phase. Figure 1 shows a diagram of the method.

To further improve performance, we generate multiple candidate prompts for each question-rubric group, varying the prompt formulation (see Appendix B), the thinking budget (medium or high), and the use of synthetic examples (explained in Section 2.2). For each group, we then retain the prompt that performs best on the Trial set. Table 1

reports all evaluated variants and their corresponding weighted F1 and QWK scores.

2.1 Meta-prompt generation

For each question-rubric group, we generate a dedicated scoring prompt in an offline step using the fixed question, its rubric, and the labeled training answers associated with that group. The training instances are provided as a JSON object containing each answer and its gold label. An LLM then infers the scoring criteria that best characterize the group and produces a reusable prompt template for evaluating unseen answers. The meta-prompt encourages the model to capture the semantic distinctions underlying the rubric instead of relying on superficial lexical patterns. The resulting template includes placeholders for the *question*, *rubric*, and *answer*, and instructs the downstream model to output only the final *label*. This process yields one candidate scoring prompt per group, which can later be used directly or compared with alternative prompt variants. The complete prompt template is provided in Appendix A.

2.2 Synthetic data

As part of a strategy to normalize the data imbalance within each question, we implemented a synthetic data generation module.

The pipeline is implemented as a state graph (using LangGraph² and LangChain³). Given a question, a student answer, and the rubric description only for the target label, the LLM is prompted to produce a new student answer matching that

²<https://github.com/langchain-ai/langgraph>

³<https://github.com/langchain-ai/langchain>

Approach	Weighted F1	QWK
Baselines		
Prompting Llama 3.1 8B Instruct	0.610	0.050
Prompting Gemini 3 Flash	0.820	0.563
Classical Machine Learning		
SVM + TF-IDF (2-10)grams	0.815	0.545
Tuning Llama 3.1 8B		
Fine-tuning LoRA	0.808	0.523
Prompt tuning	0.824	0.577
Prompt tuning with synthetic data	0.826	0.583
Ensemble prompt tuning with Gemini baseline	0.832	0.595
Meta-prompting		
Meta-prompt thinking medium	0.851	0.649
Meta-prompt thinking high	0.845	0.635
Meta-prompt thinking high with synthetic data	0.856	0.659
Meta-prompt thinking high (different prompt)	0.859	0.661
Meta-prompt best variant per group	0.892	0.743

Table 1: Results on the Trial set. In bold, the best result for each metric.

quality level. The prompt instructs the model to role-play as a real student, explicitly forbidding meta-commentary, references to grading criteria, or verbatim copying of the reference answer. Additionally, all previously generated answers for the same question are included in the prompt, instructing the model to produce a substantially different response in terms of wording, structure, and reasoning.

This is followed by a noise injection module to add grammatical errors and colloquialisms that mimic real-world writing. Finally, each response is validated against the full rubric and a conditional retry system that discards inconsistent examples to ensure the quality of the synthetic dataset.

We applied this method to the 25 questions with the greatest imbalance between correct and incorrect answers in the Train set, producing the number of samples necessary to reduce the gap between the two categories by half.

3 Other approaches

Apart from meta-prompting, we explored other approaches ranging from classic machine learning algorithms to fine-tuning open LLMs.

3.1 Role-playing prompting

Vaguely inspired in contemporary works that report Role-playing prompting may change the expected behavior of an LLM (Wang et al., 2025b,a; Luo et al., 2026), we wanted to give a try on a prompt

that described the classification task differently.

Therefore, we prompted Gemini 3 flash to act as isolated teachers in a *distant planet*. The API was called three times, always giving as an input the original question, the student’s answer, and the rubric. The first call prompts the model to act as a teacher that focus on the positive side of the student’s answer, while the second one focuses on the negative aspects. The outputs of those calls are two reviews given as an input to the third teacher (a new API call), acting as a meta-reviewer that has to make the final decision. Given how expensive it was to run this experiment (it takes 3 API calls to classify each instance), we only submitted its predictions for the *2 way unseen questions* track, without evaluating on the Trial set. Appendix C includes the prompts used for this approach.

3.2 Prompt tuning and LoRA fine-tuning

Following our approach in previous shared tasks, where we fine-tuned small LLMs (Baladón et al., 2023; Sastre et al., 2024; Góngora et al., 2025), we also explored this strategy using Llama 3.1 8B Instruct (Grattafiori et al., 2024).

In the unseen answers tracks, the task can be naturally decomposed into 72 question-rubric groups, each associated with its own subset of training examples. A straightforward approach would therefore be to fine-tune one model per group. However, performing 72 separate fine-tuning runs, even with a parameter-efficient method such as Low-Rank

Submission	QWK	WF1	#
Track 1 - Unseen Answers 3-Way			
Meta-prompt thinking medium	0.729	0.728	30/44
Meta-prompt thinking high (different prompt)	0.696	0.702	35/44
Track 3 - Unseen Answers 2-Way			
Baseline prompting Gemini 3 Flash	0.598	0.835	37/51
SVM + TF-IDF (2-10)grams	0.520	0.806	43/51
Prompt tuning with synthetic data	0.492	0.786	44/51
Ensemble prompt tuning with Gemini baseline	0.537	0.809	42/51
Meta-prompt thinking medium	0.654	0.853	25/51
Meta-prompt best variant per group	0.674	0.863	14/51
Track 4 - Unseen Questions 2-Way			
Baseline prompting Gemini 3 Flash	0.490	0.789	12/39
SVM + TF-IDF (2-10)grams	0.341	0.736	38/39
Roleplaying Gemini 3 Flash	0.432	0.766	28/39

Table 2: Results for the three tracks evaluated. The QWK metric is used as the primary reference. The "#" column indicates the ranking position out of all submissions.

Adaptation (LoRA) (Hu et al., 2022), would be computationally expensive and time-consuming. In addition, the amount of training data available for each group is often limited and may be highly imbalanced.

For this reason, we explored Prompt Tuning (Lester et al., 2021), in which the base model is kept frozen and only a small set of input embeddings, known as a soft prompt, is learned. Recent work on Concept Tokens (Sastre and Rosá, 2026) has shown that, in Llama 3.1 8B, learning a single embedding using a definitional corpus can be sufficient to steer model behavior. Inspired by this idea, we learned one embedding per question-rubric group. Each embedding was trained using the corresponding training examples, augmented with synthetic data (see Section 2.2), while the Trial split was used for early stopping. At inference time, given a new answer, we construct a prompt containing the question together with the learned embedding associated with its question-rubric group.

We compared this method against a single LoRA fine-tuning run over the full Train set. As shown in Table 1, prompt tuning achieved better performance on the Trial set than this global LoRA baseline. Both adaptation methods substantially outperformed prompting alone with the same small open-source model, which obtained a QWK of 0.050, whereas prompt tuning reached 0.583. This also highlights the large gap between prompting small open-source LLMs and prompting strong propri-

etary models: under the same general prompting setup, Gemini 3 Flash achieved a QWK of 0.56.

3.3 Support Vector Machines

Since this was ultimately a classification task, we wanted to attempt it using at least one classic machine learning algorithm. We began with the premise that the scores might be inferable from the student’s writing alone, without the need for additional context. While we anticipated that this model would not compete with neural architectures, following our previous work in BEA shared tasks (Sastre et al., 2024; Góngora et al., 2025), we consider it worthwhile to quantify the widening performance gap relative to modern transformer-based models.

Using Scikit-learn (Pedregosa et al., 2011), we conducted several preliminary experiments. We tested a wide range of statistical classifiers—including Naive Bayes, Random Forest, and Support Vector Machines—alongside various bag-of-words configurations for text representation at both word and character levels. The best configuration we found when evaluating on the Trial set was feeding a sigmoid-kernel SVC⁴, with the Train set represented by a TF-IDF vectorizer⁵ using character-level n-grams in a [2, 10] range. We trained and submitted a model using this config-

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

⁵https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

uration for the *2-way unseen answers* track and another one for the *2-way unseen questions* track.

4 Results

Table 2 summarizes the performance of our submitted systems across the three shared-task tracks in which we participated. Our main focus was Track 3 (two-way unseen answers), for which we submitted the Gemini 3 Flash prompting baseline, the SVM model, the prompt tuning approach based on Llama 3.1 8B, an ensemble of the baseline and prompt tuning methods, and several variants of the meta-prompting technique. The ensemble combines the baseline and prompt-tuning systems by selecting, for each question–rubric group, the method that achieved the best performance on the Trial set.

For Track 4 (two-way unseen questions), where the questions and rubrics are not available in advance, we submitted the same Gemini 3 Flash baseline, the SVM model, and the role-playing method. The remaining approaches depend on prior access to the question–rubric groups and therefore are not applicable in this setting. Finally, for Track 1 (three-way unseen answers), we submitted only variants of the meta-prompting approach, as this method yielded the strongest results on Track 3.

The only method that improved upon the prompting baseline was the meta-prompting approach. This result is likely due to the fact that the baseline relies on Gemini 3 Flash, a much larger and more recent model than the open model used in other experiments, Llama 3.1 8B. The strength of this baseline is further supported by its ranking in the top third of submissions on Track 4.

Meta-prompting and the role-playing strategy are the only other methods that use the same base LLM, making them the most directly comparable to the baseline. Among the Gemini 3 Flash-based approaches, meta-prompting achieved the strongest performance, improving over the baseline by nearly 8 points on QWK and ranking 14th out of 51 submissions overall. At the team level, this placed us 4th out of 9 teams that submitted to Track 3.

For Track 4, the baseline was our strongest submission, outperforming the role-playing strategy and placing us 4th out of 8 teams. Finally, we placed 6th out of 8 teams on Track 1 with the meta-prompting submission. It is worth noting that we did not tune the method on the Trial set for the three-way setting, and instead reused the

same meta-prompting configuration developed for Track 3, which may partly explain the lower performance on this track.

Regarding the Llama 3.1 8B prompt tuning method, we observe a substantial drop in performance from the Trial set to the final Test set. A likely explanation is that the Trial set was used for early stopping, which may have caused the learned embeddings to overfit to those examples. This issue is further compounded by the fact that many question–rubric groups are both small and highly imbalanced, with some groups containing examples from only a single class in the Trial split.

5 Conclusions

In this paper, we presented the RETUYT-INCO participation in the BEA 2026 shared task, “Rubric-based Short Answer Scoring for German”. Our team tried different approaches across Track 1 (Unseen answers 3-way), Track 3 (Unseen answers 2-way), and Track 4 (Unseen questions 2-way).

A central part of our experimentation involved **meta-prompting**, where an LLM generated scoring prompts based on available training instances; these prompts were then used as input for a subsequent LLM inference to classify unseen texts. Alongside this technique, we evaluated the performance of traditional machine learning models and the fine-tuning of open-source LLMs.

The official results placed our systems in the mid-table across the three tracks. In Track 1, we obtained a Quadratic Weighted Kappa (QWK) of 0.729, placing 6th out of 8 participants. Our performance in the two-way tracks was notably robust, securing 4th place out of 9 in Track 3 with a QWK of 0.674, and 4th place out of 8 in Track 4 with a QWK of 0.490.

Future work should first explore how to better adapt the meta-prompting strategy to the three-way setting, where our submitted results were still preliminary. Beyond this shared task, the approach should also be evaluated in other rubric-based scoring datasets, as well as in related educational NLP tasks, to assess its generalization and robustness. Finally, our experiments highlight the substantial gap between large proprietary models and smaller open-source LLMs. Narrowing this gap remains an important direction, both to reduce dependency on closed systems and to make high-performing educational NLP tools more accessible and cost-effective.

Limitations

Many of our experiments rely on closed proprietary models that are accessible only through provider APIs. Although these models are relevant from a research perspective due to their strong performance, this dependence limits the accessibility and applicability of the proposed methods in real educational contexts. This is especially important in settings such as rural schools, where internet connectivity, privacy constraints, and cost considerations may restrict the use of external APIs. Moreover, the meta-prompting and role-playing approaches require multiple LLM calls and consume a large number of tokens, increasing both cost and latency.

Our experiments with Llama 3.1 8B were conducted using a single Google Colab Pro subscription, which limited our ability to tune hyperparameters and perform ablation studies. Consequently, these results should be viewed as preliminary rather than fully optimized. Regarding the generated synthetic data, we did not conduct a human evaluation of whether the generated samples were of good quality. We focused on the empirical side of this method, hence prioritizing seeing an improvement in the results. Additionally, we did not evaluate how the *distant planet* setting in the role-playing approach impacted the performance of the method.

Finally, we explored the three-way setting less extensively than the two-way configurations. The submitted systems for this track were mostly adaptations of methods developed for the two-way unseen answers setting, rather than approaches specifically optimized for three-way scoring. Given the stronger results obtained in the two-way setting after task-specific tuning, we believe that better-adapted versions of our methods could substantially improve performance in the three-way setting.

Ethics

Our research uses state-of-the-art, closed-source models. These models are proprietary and operate via external APIs, so they may not be suitable for processing sensitive data, particularly involving minors. The lack of transparency regarding data handling and privacy for minors presents a significant risk. We believe it should always be a priority, when using these solutions in real-world educational deployments, to use local, private infrastructure.

Furthermore, we acknowledge the substantial environmental impact associated with the deployment

of these large-scale models. The energy consumption and resulting carbon footprint of state-of-the-art closed models are significant. Given that the work in this paper is a relatively constrained classification task, we emphasize that these state-of-the-art models may be computationally excessive. We encourage the community to explore smaller, specialized models that can provide a more sustainable and resource-efficient alternatives.

Acknowledgments

This paper was funded by *Agencia Nacional de Investigación e Innovación* (ANII, Uruguay), Project No. *FSED_2_2023_1_179355*. Additionally, this research is with support from Google.org and the Google Cloud Research Credits program for the Gemini Academic Program.

References

- Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziemis, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2026. [GEPA: Reflective prompt evolution can outperform reinforcement learning](#). In *The Fourteenth International Conference on Learning Representations*.
- Xiaoyu Bai and Manfred Stede. 2023. [A Survey of Current Machine Learning Approaches to Student Free-Text Evaluation for Intelligent Tutoring](#). *International Journal of Artificial Intelligence in Education*, 33(4):992–1030.
- Alexis Baladón, Ignacio Sastre, Luis Chiruzzo, and Aiala Rosá. 2023. [RETUYT-InCo at BEA 2023 shared task: Tuning open-source LLMs for generating teacher responses](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 756–765, Toronto, Canada. Association for Computational Linguistics.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. [The Eras and Trends of Automatic Short Answer Grading](#). *International Journal of Artificial Intelligence in Education*, 25(1):60–117.
- Luis Chiruzzo, Laura Musto, Santiago Gongora, Brian Carpenter, Juan Filevich, and Aiala Rosa. 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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- 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.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berrit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, and Hendrik Drachler. 2026. Report on the bea 2026 shared task on rubric-based short answer scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Santiago Góngora, Ignacio Sastre, Santiago Robaina, Ignacio Remersaro, Luis Chiruzzo, and Aiala Rosá. 2025. [RETUYT-INCO at BEA 2025 shared task: How far can lightweight models go in AI-powered tutor evaluation?](#) In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1135–1144, Vienna, Austria. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2022. [Instruction induction: From few examples to natural language task descriptions](#). *Preprint*, arXiv:2205.10782.
- 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*.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xuan Luo, Yue Wang, Zefeng He, Geng Tu, Jing Li, and Ruifeng Xu. 2026. [A simple and efficient learning-style prompting for LLM jailbreaking](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 2389–2406, Rabat, Morocco. Association for Computational Linguistics.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- Aiala Rosá, Santiago Góngora, Juan Pablo Filevich, Ignacio Sastre, Laura Musto, Brian Carpenter, and Luis Chiruzzo. 2025. [A platform for generating educational activities to teach english as a second language](#). *Preprint*, arXiv:2504.20251.
- 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)*, pages 618–626, Mexico City, Mexico. Association for Computational Linguistics.
- Ignacio Sastre and Aiala Rosá. 2026. Concept Tokens: Learning behavioral embeddings through concept definitions. In *Findings of the Association for Computational Linguistics: ACL 2026*. Association for Computational Linguistics.
- Yi Wang, Fenghua Weng, Sibe Yang, Zhan Qin, Minlie Huang, and Wenjie Wang. 2025a. [DELMAN: Dynamic defense against large language model jailbreaking with model editing](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11465–11481, Vienna, Austria. Association for Computational Linguistics.
- Zhenhua Wang, Wei Xie, Shuoyoucheng Ma, Enze Wang, and Baosheng Wang. 2025b. [Evading llms’ safety boundary with adaptive role-play jailbreaking](#). *Electronics*, 14(24).
- Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. [Large language models are human-level prompt engineers](#). In *The Eleventh International Conference on Learning Representations*.

A Baseline prompt

This Appendix presents the prompt used for the prompting baselines.

Baseline prompt

You are an expert educational rater for
 ↪ rubric-based short-answer scoring.

Your task is to assign a binary score to a
 ↪ student's answer using the question and
 ↪ the rubric.

Label set for this 2-way setting:
 - Output "Correct" only if the student's
 ↪ answer satisfies the rubric for
 ↪ "Correct".
 - Output "Incorrect" otherwise.
 - Answers that match "Partially Correct"
 ↪ must be labeled "Incorrect".

Decision procedure:

1. Read the question, the student answer,
→ and all rubric levels.
2. Identify the full set of meaning
→ requirements for the rubric label
→ "Correct".
3. Use the question only as context to
→ interpret the student's wording.
4. Do not use outside knowledge to add
→ content that is not stated or clearly
→ implied by the student's answer.
5. Accept paraphrases, synonyms, different
→ wording, and short or fragmentary
→ answers if their meaning clearly matches
→ the rubric.
6. Output "Correct" only if all requirements
→ for a fully correct answer are present
→ and unambiguous.
7. Output "Incorrect" if any required
→ element is missing, only partially
→ present, too vague to verify, off-topic,
→ self-contradictory, nonsensical, or
→ incompatible with the rubric.
8. If the rubric allows multiple alternative
→ ways to be fully correct, any one
→ complete valid alternative is sufficient.
9. Ignore spelling and grammar errors unless
→ they make the meaning unclear.
10. Ignore extra details unless they
→ contradict the required content or make
→ the answer incompatible with the rubric.
11. For multi-part requirements, all
→ required parts must be present unless
→ the rubric explicitly states otherwise.
12. Do not output any explanation.

The question, answer, and rubric may be in
→ German. Score based on meaning, not
→ language quality.

Input:

```
<Question>
{question}
</Question>
```

```
<StudentAnswer>
{answer_to_classify}
</StudentAnswer>
```

```
<Rubric>
<Incorrect>
{rubric_incorrect}
</Incorrect>
<PartiallyCorrect>
{rubric_partially_correct}
</PartiallyCorrect>
<Correct>
{rubric_correct}
</Correct>
</Rubric>
```

Return exactly one word and nothing else:
Correct
or
Incorrect

B Meta-prompting phase 1 prompts

This Appendix presents the prompts utilized in the Metaprompting approach, the second text is an extension from the metaprompt, that is appended before the return criteria.

Meta-prompt

You are an expert in educational assessment
→ and prompt design.

Given a question, a rubric, and labeled
→ training answers for one fixed
→ question/rubric group, generate a
→ reusable prompt for scoring a NEW answer
→ for that same group.
The possible labels are Correct and
→ Incorrect (Partially Correct are
→ classified as Incorrect).

The generated prompt must:

- be specific to this question/rubric group,
- reflect the rubric and the training
→ signals,
- use training examples for illustrating the
→ classification instructions,
- instruct the model to output only the
→ final label and nothing else,
- contain exactly these placeholders:
 - {question}
 - {answer_to_classify}

Do not:

- Repeat the rubric as-is: the idea is for
→ you to reflect on the training examples
→ and generate better criteria,
- Add examples as few-shot: use the
→ examples to explain the task and
→ classification criteria, and to
→ illustrate your explanations.

Return only the generated prompt template as
→ plain text.

Do not include any explanation or extra text.

Input:

```
<Question>
{question}
</Question>
```

```
<Rubric>
<Incorrect>
{rubric_incorrect}
</Incorrect>
<PartiallyCorrect>
{rubric_partially_correct}
</PartiallyCorrect>
<Correct>
{rubric_correct}
</Correct>
</Rubric>
```

```
<TrainingExamples>
{training_examples}
</TrainingExamples>
```

Meta-prompt extension

The expected prompt should be sufficiently
→ complex, nuanced, and comprehensive so
→ that, if you were given only your own
→ prompt, you would be fully confident in
→ your ability to correctly classify all
→ training examples, without overfitting
→ to the specific examples provided. Do
→ not optimize for brevity at the expense
→ of completeness. It is perfectly
→ acceptable for the prompt to be long if
→ that is necessary to capture all
→ relevant distinctions, edge cases, and
→ decision criteria required for perfect
→ classification. Ask yourself: does this
→ prompt contain all the information needed
→ to classify every example correctly?

Your task is to generate a three-paragraph
→ discussion on reasons to consider that
→ answer as Correct or Incorrect. In order
→ to provide a framework, you can consider
→ the answer is Correct or Incorrect
→ according to the following criteria:

- The answer is Incorrect if: {incorrect}
- The answer is Correct if: {correct}

In your discussion try to focus on the
→ NEGATIVE aspect of the answer: why would
→ you consider it as Incorrect? Write your
→ review in English.

C Roleplaying prompts

This Appendix presents the prompts used in the *Role-playing prompting* approach, described in Section 3.1.

Positive reviewer

You are a teacher in a distant planet
→ correcting homework in German. You do not
→ have the possibility of communicating
→ with any of your colleagues.
Consider the following question in German
→ written by some teacher in your solar
→ system: '{question}'. A high-school
→ student wrote this answer: '{answer}'.

Your task is to generate a three-paragraph
→ discussion on reasons to consider that
→ answer as Correct or Incorrect. In order
→ to provide a framework, you can consider
→ the answer is Correct or Incorrect
→ according to the following criteria:
- The answer is Incorrect if: {incorrect}
- The answer is Correct if: {correct}

In your discussion try to focus on the
→ POSITIVE aspects of the answer: why
→ would you consider it as Correct? Write
→ your review in English.

Metareviewer

You are a teacher in a distant planet. You
→ have received two opinions from
→ colleagues in your solar system about
→ the following question-answer pair in a
→ high-school test:
- Teacher's question: '{question}'.
- High-school student's answer: '{answer}'.

On one hand, the first colleague says:
→ '{positive_review}'.
On the other hand, the second colleague says:
→ '{negative_review}'.

Your task is to have the final word.
→ Considering that the answer was written
→ by a high-school student, say if it is
→ 'Correct' or 'Incorrect', according to
→ the words of your colleagues and the
→ following criteria:
- The answer is Incorrect if: {incorrect}
- The answer is Correct if: {correct}

Write a brief essay to support your decision,
→ and then end it saying if it is
→ 'Correct' or 'Incorrect' between two #
→ (e.g. #Correct#).

Negative reviewer

You are a teacher in a distant planet
→ correcting homeworks in German. You do
→ not have the possibility of communicating
→ with any of your colleagues.
Consider the following question in German
→ written by some teacher in your solar
→ system: '{question}'. A high-school
→ student wrote this answer: '{answer}'.

SDPA at BEA 2026 Shared Task 2: Efficient LLM Fine-Tuning for Rubric-based Short Answer Scoring

Zhexiong Liu^{1*}, Jing Zhang^{2*}

¹University of Pittsburgh, Pittsburgh, Pennsylvania 15260 USA

²Emory University, Atlanta, Georgia 30322 USA

zhexiong@cs.pitt.edu, jingz1@alumni.emory.edu

Abstract

Automated short-answer scoring (ASA) is an important yet challenging task in educational assessment as it aims to evaluate open-ended student responses against predefined scoring rubrics that are often interrelated. Although large language models (LLMs) have demonstrated impressive capabilities in text understanding and reasoning, their application to ASA has primarily focused on prompt-based inference, largely due to the limited availability of annotated data required for effective model training. In this work, we investigate parameter-efficient fine-tuning strategies for LLMs using ASA annotations in German. Our experiments show that fine-tuned LLMs consistently outperform both prompt-based and ensemble-based language models, suggesting domain-adaptive LLM fine-tuning is more effective than prompting alone for ASA.

1 Introduction

Student responses to open-ended domain-specific questions are often regarded as important indicators of their acquaintance with teacher-instructed content (Berliner and Rosenshine, 2017). Faithful assessment of their responses is essential for determining how well students have internalized curricular materials. In practice, however, manually evaluating short-answer responses remains dominant in classrooms, and it is both time-consuming and cognitively intensive since the responses are often diverse and a single question may accept multiple answers from different conceptual perspectives, which makes the manual scoring exceedingly difficult to scale. For example, Figure 1 shows three student answers scored as correct, partially correct, and incorrect based on highly interrelated rubrics. These challenges have thus motivated growing interest in the development of automated short-answer scoring (ASA) systems that can provide

Question: Assess the rate of spoilage when the apples are cut
(Note: if you would like to include technical information on the chemical processes of spoilage, you can research and include it, citing your sources).

- I. Correct Answer:** The smaller the apples are cut, the larger their surface area that can react. The larger the surface area, the more collisions and therefore more effective collisions can occur, and the faster the reaction proceeds.
- II. Partially Correct Answer:** The larger the surface area of the apple, the more substances can react on the surface, e.g., oxidize (i.e., turn brown).
- III. Incorrect Answer:** Apples spoil faster at higher temperatures because the reaction rate is increased.

- I. Correct Rubric:** The students comprehensively assess the perishability of the apples based on the collision model.
- II. Partially Correct Rubric:** The students partially assess the perishability of the apples based on the collision model.
- III. Incorrect Rubric:** The students do not assess the perishability of the apples based on the collision model.

Figure 1: A short-answer scoring example where a single question could receive correct, partially correct, and incorrect answers based on the given rubrics. The original example is in German (Gombert et al., 2026).

rubric-aligned assessments with minimal human burdens. Early automated methods focused on either lexical and semantic text matches (Ramachandran et al., 2015), or training Transformer-based S-BERT (Bexte et al., 2022). Although these models achieve reasonable performance, they have limited capability to align complex student responses with nuanced rubrics, particularly when training data are scarce. Recent large language models (LLMs) have been used to solve several challenging problems in the educational domain, e.g., essay revision (Li et al., 2024), propositional logic (Liu et al., 2024b), and mathematical geometry (Zhang et al., 2024). However, existing LLM approaches to ASA have predominantly relied on prompt methods (Henkel et al., 2024; Ferreira Mello et al., 2025), which tend to underperform when several rubrics are interrelated, e.g., partially correct versus correct rubrics. The potential of fine-tuned LLMs for ASA, particularly under low-resource conditions, therefore

*These authors contributed equally to this work

warrants further exploration. In this study in particular, we investigate parameter-efficient fine-tuning (PEFT) strategies to align LLMs with the linguistic features and rubric-matching requirements of short-answer scoring tasks.

We present our system submissions to the ASA shared task co-located at the Building Educational Applications (BEA) workshop (Gombert et al., 2026). The shared task aims to contribute to non-English benchmark datasets with four submission tracks described by two dimensions: whether the answers or questions are unseen at test time, and whether scoring follows a three-way (correct, partially correct, incorrect) or two-way (correct, incorrect) scheme. The two-way tracks collapse partially correct responses into the incorrect class, thereby posing an additional challenge for interpreting interrelated rubrics. We first establish a strong baseline using small pre-trained language models (Liu et al., 2019; Devlin et al., 2019; Chan et al., 2020), and then investigate an ensemble strategy that integrates small language models with LLMs to exploit their complementary strengths. In addition, we investigate PEFT (Hu et al., 2022; Liu et al., 2024a) strategies for LLM fine-tuning to assess their task-specific adaptation under low-resource conditions. Our system achieves second place on the binary unseen-question track and third place on the remaining three tracks. These findings demonstrate that LLMs can be effectively fine-tuned for ASA even with limited labels, providing a practical solution for low-resource educational assessment.

2 Related Work

Small Language Models. In the literature, ASA mostly relied on hand-crafted features for similarity measurement between answers and rubrics, e.g., semantic-based dependency graph alignment (Mohler et al., 2011), machine learning-based methods (Burrows et al., 2015), and several scoring systems described in the SemEval-2013 Task 7 shared task (Dzikovska et al., 2013). In addition, pre-trained Transformer encoders have substantially advanced ASA tasks (Haller et al., 2022), e.g., domain-adaptive BERT (Devlin et al., 2019) trained with augmenting data from the domain-specific textbooks (Sung et al., 2019), SBERT (Reimers and Gurevych, 2019) based on transfer learning and data augmentation (Bonthu et al., 2023). Despite these advances, small encoder models have limited capacity to handle nu-

anced open-ended responses and often struggle to distinguish between incorrect and partially correct answers. This limitation has motivated increasing interest in leveraging LLMs, which offer stronger semantic understanding and reasoning capabilities.

Large Language Models. Recent work has applied large language models (LLMs) to ASA scoring. Henkel et al. (2024) suggest that LLMs with few-shot prompts achieve human scoring in K-12 education, while Chamieh et al. (2024) and Ferreira Mello et al. (2025) indicate that prompt-based LLMs are, in contrast, less competitive than fine-tuned BERT approaches, but LLMs show promising capabilities when configured with optimized prompts such as few-shot examples and clear instructions. These observations are also presented while assessing student explanations, where fine-tuned FLAN-T5 (Chung et al., 2024) outperforms LLMs in-context learning (Carpenter et al., 2024) as well as reasoning propositional logic, where a BERT-based model outperforms Llama-based LLMs (Liu et al., 2024b). Thus, these findings reveal the limited domain generalizability of LLM prompting methods and have thus motivated the development of more task-adaptive LLM solutions. More recent ASA work investigates a QLoRA-based cross-prompt fine-tuning method (Funayama et al., 2025) and a LLM fine-tuned framework with generated feedback (Aggarwal et al., 2025). However, rubric-aware LLM fine-tuning with limited data, particularly in non-English settings (Padó et al., 2024), remains underexplored, which motivates our study.

3 Dataset

The ASA scoring task is based on a novel dataset, ALICE-LP-1.0 (Gombert et al., 2026), which contains a German-language corpus of mid- and high-school level answers to questions across four STEM domains: physics, mathematics, biology, and chemistry. Students answered short-answer items embedded in synchronous Moodle-based learning activities under teacher supervision. Each instance consists of a question, a student response, a textual rubric defining the scoring rubrics, and a label on a three-way scale (*incorrect*, *partially correct*, *correct*), as well as a binary version that collapses *partially correct* into *incorrect* label (see Figure 1). The training set contains 7,899 responses, the trial set used for validating the models contains 827 responses, and the evaluation set used for

leaderboard ranking is split into two subsets: 2,008 *unseen answers* responses obtained during training and 3,168 *unseen questions* responses held out from training. The evaluation design follows the protocol established by SemEval-2013 (Dzikovska et al., 2013) and SAF (Filighera et al., 2022). An example of the ASA data is shown in Figure 1.

4 Method

We address the ASA task with a batch of small and large language models from two perspectives. **Ensemble LLMs**, as described in Section 4.1, first employ small language model encoders to learn task-specific representations from labeled responses and produce calibrated predictive probability distributions of the candidate labels. Afterward, we ensemble multiple fine-tuned models, including few-shot predictions from LLMs, i.e., Claude-Opus-4.6 (Anthropic, 2026), to provide complementary scoring predictions. The final results are based on ensemble probabilities across candidate models. In addition, **Finetuned LLMs**, as described in Section 4.2, leverage LLM layers and a classifier head for fine-tuning using parameter-efficient methods, e.g., LoRA (Hu et al., 2022) and DoRA (Liu et al., 2024a). Unlike most generative LLMs that aim to generate predictive labels by learning the next-token probabilities, our fine-tuned LLMs instead learn to directly predict probabilities over the scoring labels. The two approaches are complementary as the encoder ensembles excel when training questions overlap with test questions (unseen answers), while the instruction-tuned LLM generalizes better to novel questions (unseen questions) by leveraging its strong linguistic capabilities.

4.1 Encoder-Ensemble LLMs

We preprocess the input text by concatenating the question, student answer, and rubric into a single sequence, i.e., *Question: {question} [SEP] Answer: {answer} [SEP] Rubric: {label₁}: {desc₁} | {label₂}: {desc₂} | {label₃}: {desc₃}*. This format provides the model with the full context needed to assess the answer against the rubric criteria, inspired by prior ASA scoring approaches (Sung et al., 2019). We train three Transformer-based models, ranging from monolingual German to multilingual language models:

- **GELECTRA-large** (Chan et al., 2020): A German ELECTRA model pre-trained on a large German corpus using the replaced token detec-

tion objective. It has demonstrated strong performance on GermEval14 (Benikova et al., 2014) and GermEval18 (Wiegand et al., 2018).

- **XLM-RoBERTa-large** (Conneau et al., 2020): A multilingual model pre-trained on 100 languages using masked language modeling on 2.5TB of clean CommonCrawl data.
- **BERT-base-german-cased** (Chan et al., 2020): A German BERT model pre-trained on German Wikipedia and news corpora.

We combine predictions from the trained encoder models with few-shot predictions from Claude-Opus-4.6 (Anthropic, 2026), without fine-tuning. Claude-Opus-4.6 uses a small number of training examples as demonstrations, of which the same questions are excluded for unseen answer tracks. It returns a score label based on its rubric interpretation and reasoning capabilities. The final ensemble uses soft voting with optimized weights. Each fine-tuned encoder produces a softmax probability distribution over labels, while Claude-Opus-4.6’s predictions are regarded as hard labels (one-hot). The ensemble prediction is:

$$\hat{y} = \arg \max_c \sum_{i=1}^n w_i \cdot p_i(c | x) \quad (1)$$

where w_i are model weights optimized via the Nelder-Mead simplex algorithm (Nelder and Mead, 1965) to maximize QWK on the trial set, p_i is the predicted probability for label c , and n is the number of ensembled models. Note that models that fail to learn (QWK = 0 on the trial set) are excluded from the ensemble for that track. We also explore multiple fusion ratios between the encoder ensemble and Claude-Opus-4.6 predictions (e.g., 70% encoder and 30% Claude-Opus-4.6), as well as a confidence-based selection strategy in which the encoder ensemble is applied when its maximum softmax probability exceeds a confidence threshold (e.g., 0.7), and Claude-Opus-4.6 is used for the remaining uncertain cases.

Implementation Each model is trained with random seeds to improve robustness through diversity, yielding up to 6 model variants per track. Also, all models are trained using the HuggingFace¹ with a linear classification head. The Claude-Opus-4.6 uses 0 temperature, and a maximum token of 50. BERT-based model training hyperparameters

¹<https://huggingface.co/>

Hyperparameter	Value
Batch size	16
Learning rate (encoder)	2×10^{-5}
Optimizer	AdamW
Max epochs	8
Max sequence length	512
LR scheduler	warmup (200 steps)
Loss function	CE with class weights

Table 1: Training hyperparameters for encoder models.

are summarized in Table 1. We also employ differential learning rates (Cazé and van der Meer, 2013), applying a $5\times$ multiplier to the classification head compared to the pre-trained encoder layers. The inverse-frequency class weights in the cross-entropy loss are used to address label imbalance, which is particularly important for the 2-way track where the Incorrect class comprises 71% of samples in the training set.

4.2 Instruction-Finetuned LLMs

While encoder models are effective for classification, they take ASA scoring as pattern matching rather than explicitly rubric-based reasoning. While full fine-tuning of LLMs can address this limitation, it is prohibitively expensive and requires large-scale annotated data. We thus leverage PEFT methods (Hu et al., 2023) to adapt low-resource fine-tuning while preserving their strong linguistic capabilities. Unlike ensemble-LLMs (Sec. 4.1), which are built from multiple models, fine-tuned LLMs produce a standalone scorer directly for inference. Specifically, we use Llama-3.1-8B (Grattafiori et al., 2024) as our base model, as it has strong multilingual capabilities and instruction-following performance. The model is instruction-tuned on the scoring task, using the instruction formatted following the prior Alpaca instruction template² (Taori et al., 2023).

```

### Instruction:
Score the following student response based on
the given rubric.
### Input:
Question: {question}
Student Response: {answer}
Rubric:
- Correct: {rubric_correct}
- Partially correct: {rubric_partial}
- Incorrect: {rubric_incorrect}

```

Inspired by previous PEFT work on text classification (Liu, 2025; Liu and Litman, 2025), we employ adapter-based methods for efficient LLM fine-

²The instructions used in the experiment were in German, but are translated into English for visualization.

Hyperparameter	Value
Quantization	8-bit
LoRA/DoRA rank r	32
Alpha α	64
Target modules	q, k, v, o projections
Dropout	0.05
Learning rate	1×10^{-4}
Batch size	16
Max epochs	5
Max sequence length	512
LR scheduler	warmup (100 steps)

Table 2: Hyperparameters for LLMs with LoRA/DoRA.

tuning, i.e., LoRA (Hu et al., 2022), which freezes the pre-trained model weights and injects trainable low-rank decomposition matrices into each transformer layer. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, the modified forward pass is:

$$h = W_0x + \Delta Wx = W_0x + BAx \quad (2)$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and the rank $r \ll \min(d, k)$. This reduces trainable parameters by orders of magnitude while preserving the model’s pre-trained knowledge. In addition, we also used the recent DoRA (Liu et al., 2024a), which extends LoRA by decomposing the pre-trained weight into magnitude and direction components:

$$W' = m \cdot \frac{W_0 + BA}{\|W_0 + BA\|_c} \quad (3)$$

where $m \in \mathbb{R}^{1 \times k}$ is a learnable magnitude vector and $\|\cdot\|_c$ denotes the column-wise norm. This decomposition enables independent adaptation of weight magnitude and direction, closely approximating full fine-tuning dynamics while maintaining the parameter efficiency of LoRA. DoRA has been shown to outperform LoRA on several NLP tasks (Liu et al., 2024a).

4.2.1 Implementation

Specifically, we apply 8-bit quantization, following prior work (Liu and Litman, 2026), to reduce memory consumption. We build the framework using PyTorch³ and HuggingFace, and optimize task losses using the Adam optimizer on an Nvidia A100. We fine-tune LLMs on the training set and tune hyperparameters on the trail set. At inference time, the fine-tuned Llama3.1-8B model outputs label probabilities given the instruction-formatted input. A softmax is applied to return the predicted label. The detailed hyperparameter settings are shown in Table 2.

³<https://pytorch.org/>

Method	Model	QWK		W-Precision		W-Recall		W-F1	
		UA	UQ	UA	UQ	UA	UQ	UA	UQ
Ensemble-LLMs	BERT-base-de	0.5514	0.4035	0.8136	0.7519	0.8147	0.7592	0.8141	0.7545
	GELECTRA-large	0.6298	0.4154	0.8462	0.7583	0.8456	0.7673	0.8459	0.7606
	Optimized Ensemble	0.6516	0.4362	0.8556	0.7667	0.8575	0.7747	0.8563	0.7688
	Ensemble + Claude-Opus-4.6	0.6617	0.4609	0.8598	0.7756	0.8615	0.7812	0.8604	0.7776
Finetuned-LLMs	Llama3.1-8B + LoRA	0.6822	0.5348	0.8685	0.8056	0.8655	0.8036	0.8667	0.8045
	Llama3.1-8B + DoRA	0.6786	0.5307	0.8665	0.8056	0.8660	0.8108	0.8663	0.8069

Table 3: The 2-way track test results for Unseen Answers (UA) and Unseen Questions (UQ).

Method	Model	QWK		W-Precision		W-Recall		W-F1	
		UA	UQ	UA	UQ	UA	UQ	UA	UQ
Ensemble-LLMs	BERT-base-de	0.6667	0.4603	0.6893	0.5670	0.6688	0.5401	0.6727	0.5452
	XLM-R-large	0.6656	0.4652	0.6957	0.5819	0.6802	0.5476	0.6833	0.5521
	GELECTRA-large	0.7352	0.5145	0.7451	0.6015	0.7305	0.5755	0.7337	0.5774
	Optimized Ensemble	0.7414	0.5228	0.7488	0.6119	0.7345	0.5842	0.7377	0.5866
	Ensemble + Claude-Opus-4.6	0.7566	0.5446	0.7695	0.6165	0.7524	0.5991	0.7554	0.6030
Finetuned LLMs	Llama3.1-8B + LoRA	0.7760	0.6444	0.7748	0.6338	0.7634	0.6325	0.7661	0.6330
	Llama3.1-8B + DoRA	0.7556	0.6292	0.7454	0.6440	0.7460	0.6251	0.7452	0.6300

Table 4: The 3-way track test results for Unseen Answers (UA) and Unseen Questions (UQ).

4.3 Results

Regarding ensemble-LLMs for the two-way and three-way track performance in Table 3 and Table 4, GELECTRA-large consistently achieves the strongest individual performance, benefiting from German-specific pre-training. In terms of XLM-RoBERTa-large, it exhibits training fluctuation, with some seeds failing to converge (e.g., QWK = 0), particularly on the 2-way track, thus we did not submit its performance. In contrast, GELECTRA-large outperforms the larger multilingual XLM-RoBERTa-large in three-way track. This demonstrates that monolingual pre-training yields more useful representations than model size for language-specific classification tasks, whereas XLM-RoBERTa-large exhibits high sensitivity to experimental configuration, as some runs fail to learn in our pilot study. This might be caused by a linguistic adaption from its multilingual pre-training to the German classification objective, given the relatively small training set.

Regarding LLMs for the two-way and three-way scoring, ensemble Claude-Opus-4.6’s few-shot predictions without fine-tuning on the task consistently provide a complementary strength by leveraging rubric-level reasoning. In other words, incorporating Claude-Opus-4.6’s predictions into the encoder ensemble is particularly beneficial for ambiguous cases where discriminative models produce low-confidence outputs. In contrast, the instruction fine-tuned Llama3.1-8B models substantially outperform the ensemble-based method on both the

unseen answer and unseen question tracks. Specifically, fine-tuning yields relatively larger gains on the 2-way track than on the 3-way track, suggesting that 3-way ASA scoring, which requires distinguishing partially correct from incorrect responses, is inherently more challenging. Although LoRA generally outperforms DoRA in terms of QWK, their weighted F1 scores remain comparable. Given that LoRA is also more computationally efficient, it is regarded as the preferred PEFT method. Notably, the fine-tuning approaches offer the advantage of a single scorer that evaluates responses against rubrics without relying on an ensemble of multiple models. This is particularly beneficial for the unseen questions track, where generalization to novel questions and rubrics is critical.

5 Conclusion

We presented a set of novel ensemble-based and fine-tuned LLM methods for ASA scoring in German. The encoder ensemble augmented with few-shot Claude-Opus-4.6 predictions demonstrates advantages over small language models, while the instruction-tuned Llama3.1-8B scorer exhibits superior performance in learning complex scoring patterns. Our results suggest that parameter-efficient fine-tuning enables LLMs to score diverse student responses even with limited training data. These findings offer a practical solution for deploying LLM-based ASA scoring systems in low-resource educational assessment.

Limitations

Our encoder ensemble optimization relies on the trial set for weight tuning, which may lead to overfitting on the evaluation set. In addition, the ensemble approach increases system complexity, requiring training multiple models and careful ensemble-based strategies. While fine-tuned LLMs with PEFT achieve substantial improvement, they require GPU resources, potentially limiting deployment in low-resource educational contexts. Furthermore, our experiments are conducted exclusively on German short-answer data, and the effectiveness of the proposed methods on other languages or domains remains unverified.

Ethics Statement

This work uses the ALICE-LP-1.0 dataset, which was collected and annotated for academic research purposes with appropriate institutional oversight. As the dataset contains student responses from minors in German secondary schools, we do not attempt to re-identify individual students. Our automated scoring systems are intended to assist, not replace, human graders; deployment in high-stakes assessment settings would require thorough validation and human oversight to ensure fairness across student populations. We acknowledge that both encoder models and LLMs may introduce biases during pretraining that could affect scoring equity. Additionally, LLM-generated scores may exhibit inconsistencies or errors that require careful human review prior to educational deployment. Lastly, the use of LLMs may raise practical concerns regarding cost, accessibility, and reproducibility.

References

- Dishank Aggarwal, Pritam Sil, Bhaskaran Raman, and Pushpak Bhattacharyya. 2025. “i understand why i got this grade”: Automatic short answer grading (asag) with feedback. pages 304–318.
- Anthropic. 2026. [Introducing claude opus 4.6](#).
- Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Pado. 2014. Germeval 2014 named entity recognition shared task: companion paper. In *Workshop Proceedings of the 12th edition of the KONVENS conference*, pages 104–112.
- David C Berliner and Barak Rosenshine. 2017. The acquisition of knowledge in the classroom. In *Schooling and the acquisition of knowledge*, pages 375–396. Routledge.
- 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.
- Sridevi Bonthu, S. Rama Sree, and M.H.M. Krishna Prasad. 2023. [Improving the performance of automatic short answer grading using transfer learning and augmentation](#). *Engineering Applications of Artificial Intelligence*, 123:106292.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. [The eras and trends of automatic short answer grading](#). *International Journal of Artificial Intelligence in Education*, 25:60–117.
- Dan Carpenter, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng, and James Lester. 2024. Assessing student explanations with large language models using fine-tuning and few-shot learning. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 403–413.
- Romain D Cazé and Matthijs AA van der Meer. 2013. Adaptive properties of differential learning rates for positive and negative outcomes. *Biological cybernetics*, 107(6):711–719.
- Imran Chamieh, Torsten Zesch, and Klaus Giebertmann. 2024. LLMs in short answer scoring: Limitations and promise of zero-shot and few-shot approaches. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 309–315.
- Branden Chan, Stefan Schweter, and Timo Möller. 2020. [German’s next language model](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.
- Alexis Conneau, Kartik 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*, pages 8440–8451.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

- Myroslava O 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.
- Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. 2025. [Automatic short answer grading in the llm era: Does gpt-4 with prompt engineering beat traditional models?](#) In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK '25*, page 93–103, New York, NY, USA. Association for Computing Machinery.
- Anna Filighera, Siddharth Parihar, Tim Steuer, Tobias Meuser, and Sebastian Ochs. 2022. Your answer is incorrect... would you like to know why? introducing a bilingual short answer feedback dataset. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8577–8591.
- Hiroaki Funayama, Yuichiroh Matsubayashi, Yuya Asazuma, Tomoya Mizumoto, and Kentaro Inui. 2025. Cross-prompt pre-finetuning of language models for short answer scoring. *International Journal of Artificial Intelligence in Education*, pages 1–22.
- Sebastian Gombert, Zhifan Sun, Fabian Zehner, Jannik Lossjew, Tobias Wyrwich, Berrit Katharina Czinczel, David Bednorz, Sascha Bernholt, Knut Neumann, Ute Harms, Aiso Heinze, and Hendrik Drachslar. 2026. Report on the bea 2026 shared task on rubric-based short answer scoring for german. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. 2022. Survey on automated short answer grading with deep learning: from word embeddings to transformers. *arXiv preprint arXiv:2204.03503*.
- Owen Henkel, Libby Hills, Adam Boxer, Bill Roberts, and Zach Levonian. 2024. [Can large language models make the grade? an empirical study evaluating llms ability to mark short answer questions in k-12 education.](#) In *Proceedings of the Eleventh ACM Conference on Learning @ Scale, L@S '24*, page 300–304, New York, NY, USA. Association for Computing Machinery.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shanen Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. *Proceedings of ICLR*.
- Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5254–5276.
- Tianwen Li, Zhexiong Liu, Lindsay Matsumura, Elaine Wang, Diane Litman, and Richard Correnti. 2024. [Using large language models to assess young students' writing revisions.](#) In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 365–380, Mexico City, Mexico. Association for Computational Linguistics.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. DoRA: Weight-decomposed low-rank adaptation. *Proceedings of ICML*.
- 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*.
- Zhexiong Liu. 2025. *Understanding and Generating Text Revisions With Intention-Adaptive Large Language Models*. Ph.D. thesis, University of Pittsburgh.
- Zhexiong Liu and Diane Litman. 2025. [Efficient layer-wise LLM fine-tuning for revision intention prediction.](#) In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15319–15334, Suzhou, China. Association for Computational Linguistics.
- Zhexiong Liu and Diane Litman. 2026. [Intention-adaptive LLM fine-tuning for text revision generation.](#) In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1263–1281, Rabat, Morocco. Association for Computational Linguistics.
- Zhexiong Liu, Jing Zhang, Jiaying Lu, Wenjing Ma, and Joyce C. Ho. 2024b. [Logicprpbank: A corpus for logical implication and equivalence.](#) In *Proceedings of the 2024 AAAI Conference on Artificial Intelligence*, volume 257 of *Proceedings of Machine Learning Research*, pages 57–65. PMLR.
- Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. [Learning to grade short answer questions using semantic similarity measures and dependency graph alignments.](#) In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA. Association for Computational Linguistics.

- John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.
- Ulrike Padó, Yunus Eryilmaz, and Larissa Kirschner. 2024. Short-answer grading for german: Addressing the challenges. *International Journal of Artificial Intelligence in Education*, 34(4):1321–1352.
- Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the tenth workshop on innovative use of NLP for building educational applications*, pages 97–106.
- 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.
- Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. 2019. [Pre-training BERT on domain resources for short answer grading](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6071–6075, Hong Kong, China. Association for Computational Linguistics.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*. <https://crfm.stanford.edu/2023/03/13/alpaca.html>, 3(6):7.
- Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.
- Jiaxin Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, Cheng-Lin Liu, and Yashar Moshfeghi. 2024. Geoeval: benchmark for evaluating llms and multi-modal models on geometry problem-solving. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1258–1276.

Author Index

- Adi, Ahmad, 1124
Aguinalde, Pauline, 1077
Akef, Soroosh, 851
Akram, Bitá, 598, 841
Alexandron, Giora, 432
Alhafni, Bashar, 368
Ali, Mahmoud, 702
Alve, Kamruzzaman Khan, 1041
ALZU'BI, SARH, 766
Amjad, Ahmad Imam, 951
Amjad, Maaz, 951
Andersen, Nico, 624
Anderson, Nathan, 48
Andreev, Dmitrii, 1091
Anisimova, Mariia, 141
Ariely, Moriah, 432
- Bai, Xiaoyu, 405
Baldwin, Peter, 153
Banno, Stefano, 174, 368
Barakat, Mariam, 863
Barnes, Tiffany, 819
Bednorz, David, 1179
Beigman Klebanov, Beata, 432, 546
Beinborn, Lisa, 1047
Belcher, Kate, 1224
Bernholt, Sascha, 1179
Bestgen, Yves, 985
Bexte, Marie, 1201
Bhatia, Aakanksha, 546
Bhatt, Tirth, 841
Bibi, Asifa, 1106
Bodnar, Stephen, 190
Both, Andreas, 1210
Bralin, Amir, 93
Brusilovsky, Peter, 598, 841
Butt, Sabur, 1106, 1137
Buttery, Paula, 562
- Caines, Andrew, 562, 715
Camus, Leon, 259, 624
Cao, Fanghua, 751
Castro, Francisco, 598
Ceballos Cancino, Héctor, 1137
Chamieh, Imran, 419
Chan, Sophia, 281
Chang, Yi-Ning, 1161
Chao, Fu-An, 21, 108
- Chen, Berlin, 21, 108
Chen, Xiaobin, 443, 933, 997
Chen, Yinong, 941
Chernishev, George, 463
Chiruzzo, Luis, 1113, 1235
Chiu, Shih-Hsuan, 21
Cho, Seok Hyeon, 1029
Choi, JunHyeok, 1029
Christenson, Julia, 685
Chu, Yuan, 443
Colling, Leona, 190
Cong, Longwei, 259, 624
Czajka, Mateusz, 492
Czinczel, Berrit, 1179
- Dalbo, Emma, 992
Dascalu, Mihai, 358
De Horta, Nicolás, 1235
De Kuthy Meurers, Marius, 1224
De Kuthy, Kordula, 1224
de Langis, Karin, 685
Degraeuwe, Jasper, 715
Dhall, Adit, 1010
Ding, Yuning, 1201
Domingue, Ben, 777
Dorr, Bonnie, 751
Drachsler, Hendrik, 259, 624, 1179
Drackert, Anastasia, 272
Dutulescu, Andreea, 358
Díaz, Facundo, 1235
- Eryiğit, Gülşen, 234
Esiason, Jordan, 248
Evanini, Keelan, 153
- Fan, Jing, 598
Farhana, Effat, 1
Felice, Mariano, 964
Frome, Kevin, 153
- Gaillat, Thomas, 1065
Gales, Mark, 174
Gao, Kevin, 813
Gaudeau, Gabrielle, 715
Gelbukh, Alexander, 1106
Giebertmann, Klaus, 419
Gilda, Shlok, 751
Gilson, Sam, 819

Gohr, Aron, 813
 Goldwasser, Dan, 93
 Gombert, Sebastian, 259, 624, 1179
 Goodman, Bradley, 7
 Goto, Takumi, 899
 Gowda, Thejas, 1010
 Grand-Clement, Caroline, 503
 Grover, Utkarsh, 529
 Gurin Schleifer, Abigail, 432
 Gwozdz, Jonas, 1210
 Góngora, Santiago, 1235

 Ha, Le An, 574
 Haber, Nick, 777
 Hahn, Sonja, 259, 624
 Halder, Shreyashi, 546
 Han, Sung Won, 1029
 Hardy, Michael, 651
 Harik, Polina, 574
 Harms, Ute, 1179
 Haseeb, Muhammad, 951
 Hayati, Shirley Anugrah, 685
 He, Mengyuan, 443
 He, Yulan, 292
 Heinze, Aiso, 1179
 Hellas, Arto, 598, 841
 Herygers, Aaricia, 1047
 Heslip, Stephen, 813
 Hirschberg, Julia, 34
 Hmue, Min Paing, 951
 Hoang, Andrew, 546
 Holz, Heiko, 63
 Horbach, Andrea, 221, 347, 1201
 Hossain, Abid, 1041
 Hou, Jue, 479
 Howson, Oli, 635
 Hrabal, Miroslav, 141
 Huang, Lanhua, 933
 Huang, Shuyan, 612
 Huang, Zhuojing, 1047

 Ide, Yusuke, 1161
 Issa, Elsayed, 702

 Jain, Mragisha, 841
 Jansen, Thorben, 347
 Jeong, Yeil, 248
 Ji, Sangeun, 1029

 Kafunah, Jefkine, 1065
 Kang, Dongyeop, 685

 Kanojia, Diptesh, 1016
 Kapoor, Radhika, 777
 Kashyap, Sanjna, 546
 Kathmann, Sofia, 63
 Katinskaia, Anisia, 479
 Katunin, Pavel, 1091
 Keller, Stefan, 221
 Keluskar, Aryan, 941
 Khalid, Haziq, 581
 Khallaf, Nouran, 1146
 Khare, Priyanka, 248
 Kloudová, Věra, 141
 Knight, Ryan, 48
 Knill, Kate, 174
 Kochmar, Ekaterina, 118, 368, 863
 Kopan, Artyom, 463
 Kroehne, Ulf, 259
 Kuczyński, Tomasz, 492
 Kumar, Ravi, 529
 Köller, Olaf, 347

 Laarmann-Quante, Ronja, 272
 Lade, Shirin, 635
 Lan, Andrew, 78, 612
 Latif, Tayyab, 1106
 Lawn, Marie-Amelie, 813
 Lee, Grandee, 798
 Lee, Jaewook, 78, 612
 Lee, Seung, 248
 Lee, Younghun, 93
 Leinonen, Juho, 598, 841
 Leite, Walter, 751
 Levin, Hannah, 513
 Li, Jiazheng, 292
 Li, Jiun-Ting, 21, 108
 Li, Wenbo, 819
 Li, Zhe, 1077
 Liang, Wen, 34
 Lin, Hong-Yun, 108
 Lin, Xiaomin, 529
 Liu, Shiqin, 443
 Liu, Xinman, 208
 Liu, Zhexiong, 1244
 Lo, Kyle, 48
 Lo, Tien-Hong, 21
 Loefflad, Denise, 63
 Logacheva, Evanfiya, 598
 Longwill, Benny, 546
 Lopo, Joanito, 1124
 Lossjew, Jannik, 1179
 Lucy, Li, 48

Lye, Che Yee, 798
 Lyu, Wenhan, 668
 Löber, Sarah, 443

 M., wicaksono, 1124
 Makeev, Vladislav, 463
 Manlises, Maria Monica, 914
 Mann, Stephanie, 153
 Masciolini, Arianna, 503
 Mathias, Sandeep, 1119
 Mattu, Karthik, 1010
 Maurya, Kaushal, 368
 Mayer Martins, Jonas, 1047
 McCrae, John P., 1065
 Mcglone, Desmond, 668
 McNamara, Danielle, 358
 Mee, Janet, 574
 Mehler, Alexander, 624
 Mendes, Amália, 851
 Meurers, Detmar, 63, 190, 851, 1224
 Micir, Ian, 153
 Mihaylova, Tsvetomila, 598
 Mikeska, Jamie, 546
 Min, Wookhee, 248
 Mireshghallah, Niloofar, 1
 Moore, Russell, 562
 Mora Melanchthon, Daniel, 221, 347, 1201
 Mwiinga, Nchimunya, 513

 Naguru, Arshad, 1010
 Neumann, Knut, 1179
 Nguyen, Ngoc Nhu Hao, 624
 Nguyen, Tien Dat, 118
 Nikolaev, Anton, 1091
 Nohejl, Adam, 1161
 Norouzi, Narges, 598, 841
 Novák, Michal, 141
 Nugraha, Tsamarah, 1124
 Nurfajri, Muhamad, 1124
 Nylk, Jonathan, 635

 O'Donnell, Francis, 455
 O'Driscoll, Aoife, 715
 Ong, Ethel, 914
 Orasan, Constantin, 1016
 Ozogul, Gamze, 248

 Pado, Ulrike, 1193
 Padwal, Madhura, 513
 Painter, Jordan, 1016
 Pandya, Aum, 841

 Parashar, Anubhab, 1119
 Park, Seongjin, 336
 Peh, Luke, 798
 Petukhova, Kseniia, 118
 Philippov, Vassili, 1091
 Pineda, Adrian, 1137
 Pitts, Griffin, 841
 Piłka, Tomasz, 492
 Polakova, Lucie, 141
 Polyzou, Agoritsa, 529
 Popel, Martin, 141

 Rackauckas, Zackary, 34
 Rao, Anmol, 941
 Rebello, Nobel Sanjay, 93
 Rebuschat, P, 851
 Remersaro, Ignacio, 1235
 Reynolds, Robert, 766
 Rezayi, Saed, 153, 574
 Ribeiro-Flucht, Luisa, 933
 Riera Machin, Maria, 1161
 Robaina, Santiago, 1113
 Rooein, Donya, 715
 Rosá, Aiala, 1113, 1235
 Rudzewitz, Björn, 443
 Ruiz-Primo, Maria, 777
 Runyon, Christopher, 153
 Ruppenhofer, Josef, 1201
 Ruseti, Stefan, 358

 Sakai, Yusuke, 899
 Salman, Asaf, 432
 Sastre, Ignacio, 1235
 Scarlatos, Alexander, 78, 612
 Schaller, Nils-Jonathan, 347, 1201
 Sehgal, Shubh, 1010
 Serjeant, Inigo, 813
 Shapsough, Salsabeel, 581
 Sharma, Avinash Kumar, 1024
 Sharma, Mayank, 208, 777
 Sharoff, Serge, 1146
 Shen, Qian, 751
 Sheng, Victor, 951
 Shi, Xinyu, 208
 Shin, Jinnie, 1077
 Shu, Ziqi, 651
 Sidorov, Grigori, 1106
 Siyan, Li, 34
 Skidmore, Lucy, 964
 Slutz, Emily, 668
 Smirnova, Anna, 463

Snyder, Jason, 574
 Soliar, Mariia, 190
 Solis de Ovando, Madeline, 281
 Srikanth, Shankhalika, 281
 stearns, bernardo, 1065
 Stede, Manfred, 405
 Sternik, Malte, 272
 Suh, Jennifer, 668
 Sun, Zhifan, 624, 1179
 Sung, Hakyung, 1010

 Talat, Zeerak, 715
 Third, Aisling, 1217
 Tian, Xiaoyi, 819
 Tithi, Sutapa, 819
 Truong, Sang, 777
 Türker, Tolgahan, 234

 Vajjala, Sowmya, 368
 van der Meulen, Ramira, 1
 van Duijn, Max, 1
 Vu, Anh-Duc, 479

 Wang, Qiaosi, 1
 Wang, Shuman, 651
 Wang, Yue, 798
 Watanabe, Taro, 899
 Wells, Emily, 1016
 Willis, Alistair, 635, 1217
 Willoughby, Georgina, 1016

Wong, Gee-Lyle, 292
 Woo, Ryan, 941
 Woodhead, Simon, 78
 Wu, Xuanxin, 1161
 Wu, Yiheng, 479
 Wyrwich, Tobias, 1179

 Yan, Bi-Cheng, 21, 108
 Yaneva, Victoria, 153, 455, 574
 Yangarber, Roman, 479
 Yao, Min, 751
 Yao, Ziyu, 668
 Yasir, Tahreem, 819
 Ye, Yushan, 443
 Yu, Sabrina, 281
 Yue, Murong, 668

 Zainab, Maha, 1
 Zehner, Fabian, 624, 1179
 Zesch, Torsten, 419, 1201
 Zhang, Albert, 48
 Zhang, Jing, 1244
 Zhang, Yixuan, 668
 Zhao, Runcong, 292
 Zheng, Xiaoying, 248
 Zhou, Yiyun, 455
 Zolotilin, Mikhail, 479
 Zualkernan, Imran, 581